1 PRODUCT OVERVIEW

OVERVIEW

Samsung S3C380D 16/32-bit RISC microcontroller is a cost-effective and high-performance microcontroller solution for TV applications.

Among the outstanding features of the S3C380D is its CPU core, a 16/32-bit RISC processor (ARM7TDMI) designed by Advanced RISC Machines, Ltd. The ARM7TDMI core is a low-power, general-purpose microprocessor macro-cell that was developed for use in application-specific and customer-specific integrated circuits. Its simple, elegant, and fully static design is particularly suitable for cost-sensitive and power-sensitive applications.

The S3C380D was developed using the ARM7TDMI core, CMOS standard cell, and a data path compiler. Most of the on-chip function blocks were designed using an HDL synthesizer. The S3C380D has been fully verified in the Samsung ASIC test environment.

By providing a complete set of common system peripherals, the S3C380D minimizes overall system costs and eliminates the need to configure additional components.

The integrated on-chip functions that are described in this document include:

- 4-Kbyte RAM (3008-byte (1504 × 16 bits) general register and 1088-byte (544 × 16 bits) OSD/CCD RAM)
- 128-Kbyte internal program memory
- Two 14-bit PWM modules
- Three 16-bit timers
- On screen display module
- Crystal/Ceramic oscillator or external clock can be used as the clock source
- Standby mode support: SLEEP mode
- One 8-bit basic timer and 3-bit watchdog timer
- Interrupt controller (16 interrupt sources and 2 vectors)
- Five 4-bit ADCs
- Four programmable I/O ports
- 42-pin SDIP

FEATURES

CPU

- ARM7T CPU core

Memory

- 4-Kbyte RAM (3008-byte general purpose register area + 1088-byte OSD/CCD RAM)
- 128 Kbyte internal program memory

General I/O

- Four I/O ports (25 pins total)
  (6 V O/D: 3 pins, 5 V O/D: 4 pins)

Basic timer and watchdog timer

- 8-bit counter + 3-bit counter
- Overflow signal of 8-bit counter makes a basic timer interrupt and control the oscillation warm-up time
- Overflow signal of 3-bit counter makes a system reset

Timer/Counters

- Three general purpose 16-bit timer/counters with interval timer modes

Interrupts

- 16 interrupt sources and 2 vectors
- Fast interrupt processing
- 2 interrupt shadow registers (32 bit $\times$ 2)

Pulse width modulation (PWM) module

- 14-bit PWM with 2-channel PWM counter

A/D converter

- 5-channel: 4-bit conversion resolution (flash ADC)

Remocon receiver

- FIFO 8 steps
- FIFO interrupt is full (8) step overflow

On screen display (OSD) mode

- Analog level OSD
- Halftone
- 64 character colors
- 16 different character sizes
- Graphic OSD
- S/W CCD

Oscillator frequency

- 32,768 Hz external crystal oscillator
- 1 Hz generation for real time clock
- PLL (Phase Lock Loop) controlled oscillators
- Maximum 16 MHz CPU clock

Operating temperature Range

- - 20 °C to + 85 °C

Operating Voltage Range

- 4.5 V to 5.5 V

Package Type

- 42-pin SDIP

BLOCK DIAGRAM

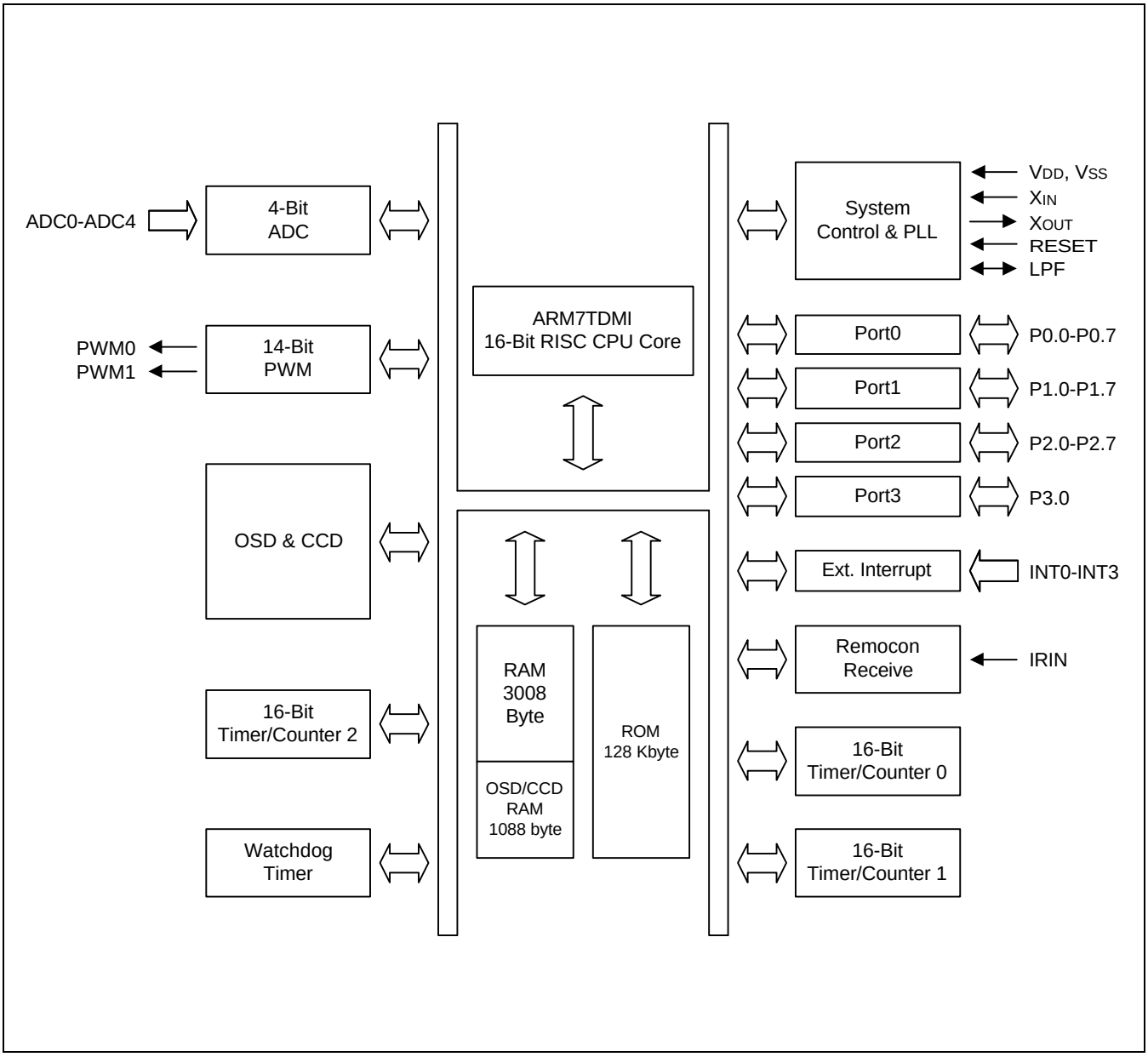


Figure 1-1. S3C380D Block Diagram

PIN ASSIGNMENTS

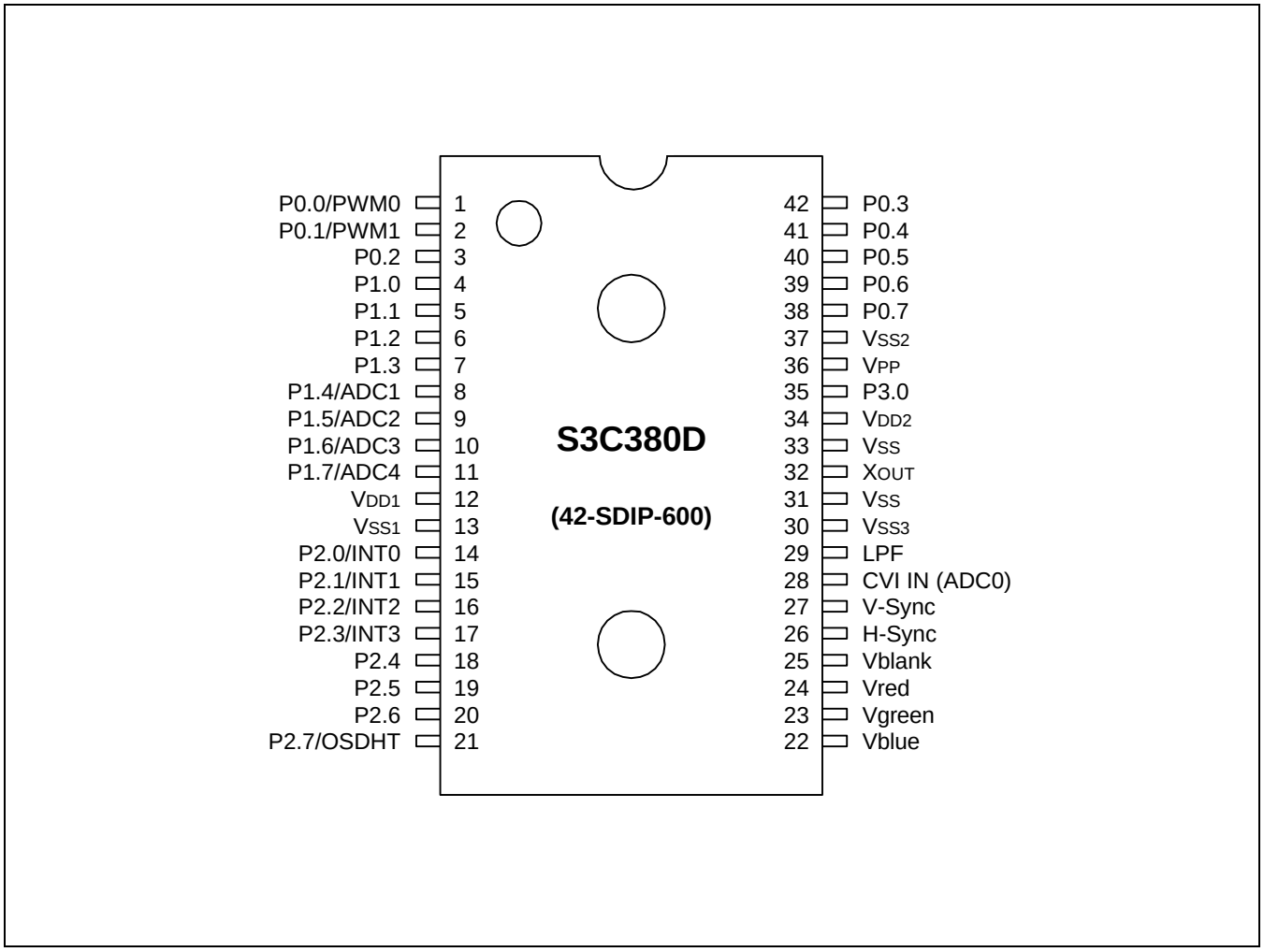


Figure 1-2. S3C380D Pin Assignments (42-SDIP)

PIN DESCRIPTIONS

Table 1-1. S3C380D Pin Descriptions

Pin Name	Pin Type	Pin Description	Circuit Type	Pin Numbers	Share Pins
P0.0	I/O	Input mode or push-pull output mode is software configurable. P0.0: PWM0 (14-bit PWM Output)	6	1	PWM0
P0.1-P0.2 P0.3		General I/O Port (3-bit), Input or n-channel open-drain output is software configurable. Pins can withstand up to 6-volt loads. An alternative function is supported. P0.1: PWM1 (14-Bit PWM Output)	3	2-3 42	PWM1
P0.4-P0.7		General I/O Port (4-bit), Input or Output mode (push-pull or n-channel open drain) is software configurable.	7	38-41	
P1.0-P1.3	I/O	Input/output mode or push-pull output mode is software configurable.	6	4-7	
P1.4-P1.7		General I/O Port (4-bit), configurable for digital input or n-channel open drain output. P1.4-P1.7 can withstand up to 5-volt loads. Multiplexed for alternative use as external inputs ADC1-ADC4.	4	8-11	ADC1-ADC4
P2.0-P2.3	I/O	General I/O Port (4-bit), input or push-pull output mode is software configurable. Multiplexed for alternative use as external interrupt inputs INT0-INT3.	2	14-17	INT0-INT3
P2.4-P2.7	I/O	Input mode or push-pull output mode is software configurable. An alternative function is supported. P2.7: OSDHT (Halftone signal output)	6	18-21	OSDHT
P3.0	I/O	Input mode or push-pull output mode is software configurable.	6	35	
PWM0	O	Output pin for 14-bit PWM0 circuit	6	1	P0.0
PWM1	O	Output pin for 14-bit PWM1 circuit	3	2	P0.1
ADC1-4	I	Input for 4-bit resolution flash A/D Converter	4	8-11	P1.4-7
INT0-INT3	I	External interrupt input pins	2	14-17	P2.0-3
OSDHT	O	Halftone control signal output for OSD	6	21	P2.7
IRIN	I	Remocon signal input Normal mode: Remocon signal input OTP Write mode: V_{PP}=12.5 V	1	36	–
CVI IN	I	Video signal input	8	28	ADC0

Table 1-1. S3C380D Pin Descriptions (Continued)

Pin Name	Pin Type	Pin Description	Circuit Type	Pin Numbers	Share Pins
RESET	I	System reset input pin	9	33	–
LPF	–	PLL filter pin	–	29	–
H-SYNC	I	H-sync input for OSD and CCD	1	26	–
V-SYNC	I	V-sync input for OSD and CCD	1	27	–
V_{blank}	O	Video blank signal output for OSD and CCD	5	25	–
V_{red}	O	Red signal output for OSD and CCD	5	24	–
V_{green}	O	Green signal output for OSD and CCD	5	23	–
V_{blue}	O	Blue signal output for OSD and CCD	5	22	–
ADC0	I	Input for 4-bit resolution flash A/D Converter (1.5V-2.0V)	8	28	CVI IN
V_{DD1}, V_{DD2} V_{SS1}, V_{SS2} V_{SS3}	–	Power supply pins	–	12, 34 13, 37 30	–
X_{IN}, X_{OUT}	I, O	System clock pins (32,768 Hz)	–	31,32	–

PIN CIRCUITS

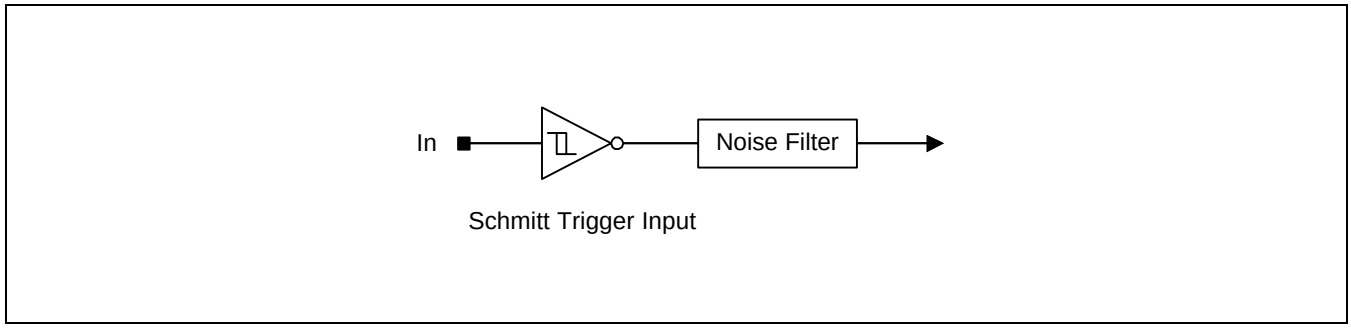


Figure 1-3. Pin Circuit Type 1 (H-Sync, V-Sync, IRIN)

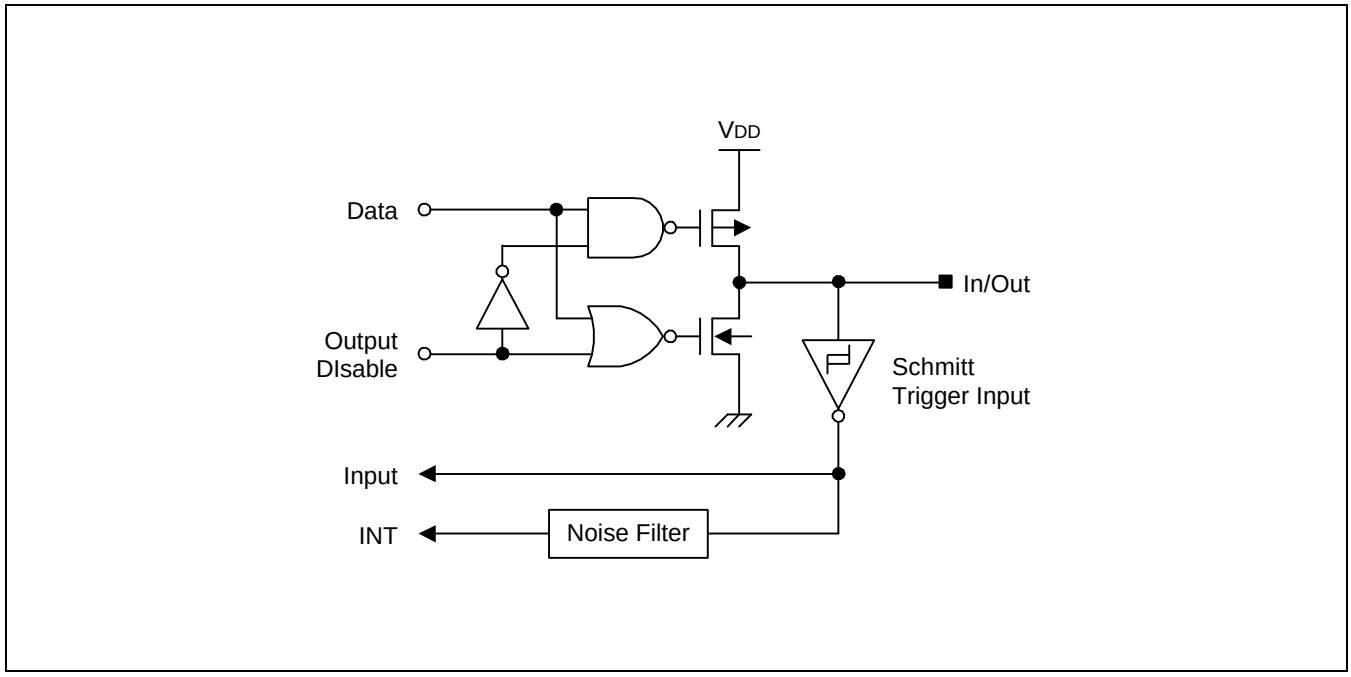
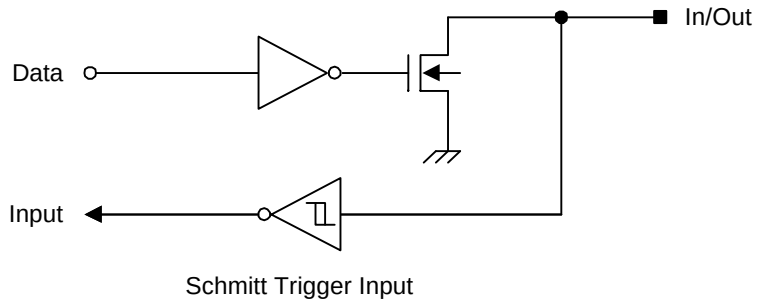
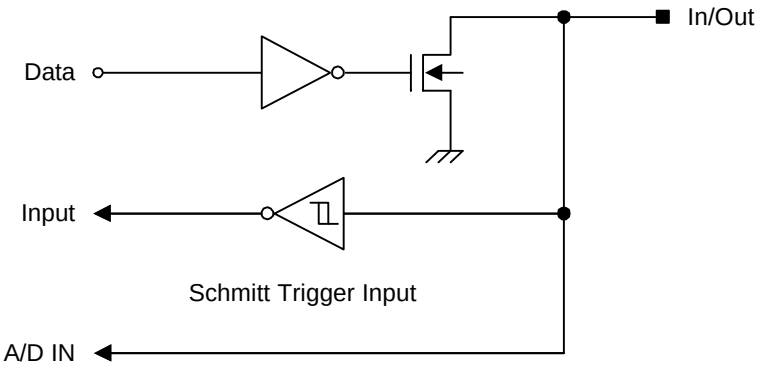


Figure 1-4. Pin Circuit Type 2 (P2.0-P2.3, INT0-INT3)



NOTE: Circuit type 3 can withstand up to 6 V loads.

Figure 1-5. Pin Circuit Type 3 (P0.1-P0.3, PWM1)



NOTE: Circuit type 4 can withstand up to 5 V loads.

Figure 1-6. Pin Circuit Type 4 (P1.4-P1.7, ADC1-ADC4)

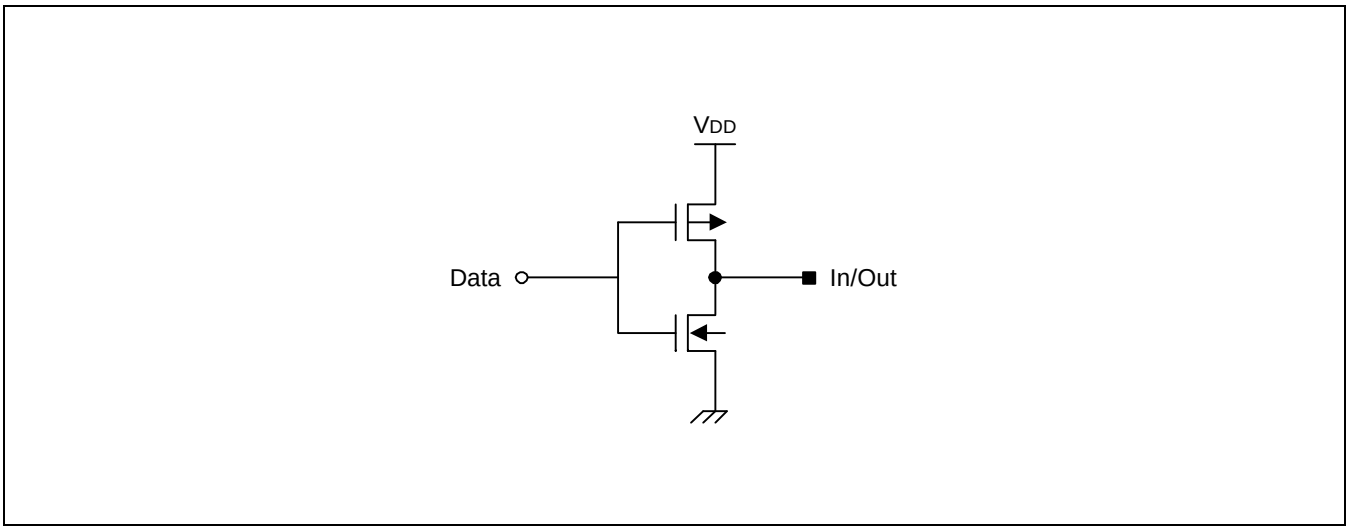


Figure 1-7. Pin Circuit Type 5 (V_{blue}, V_{green}, V_{red}, V_{blank})

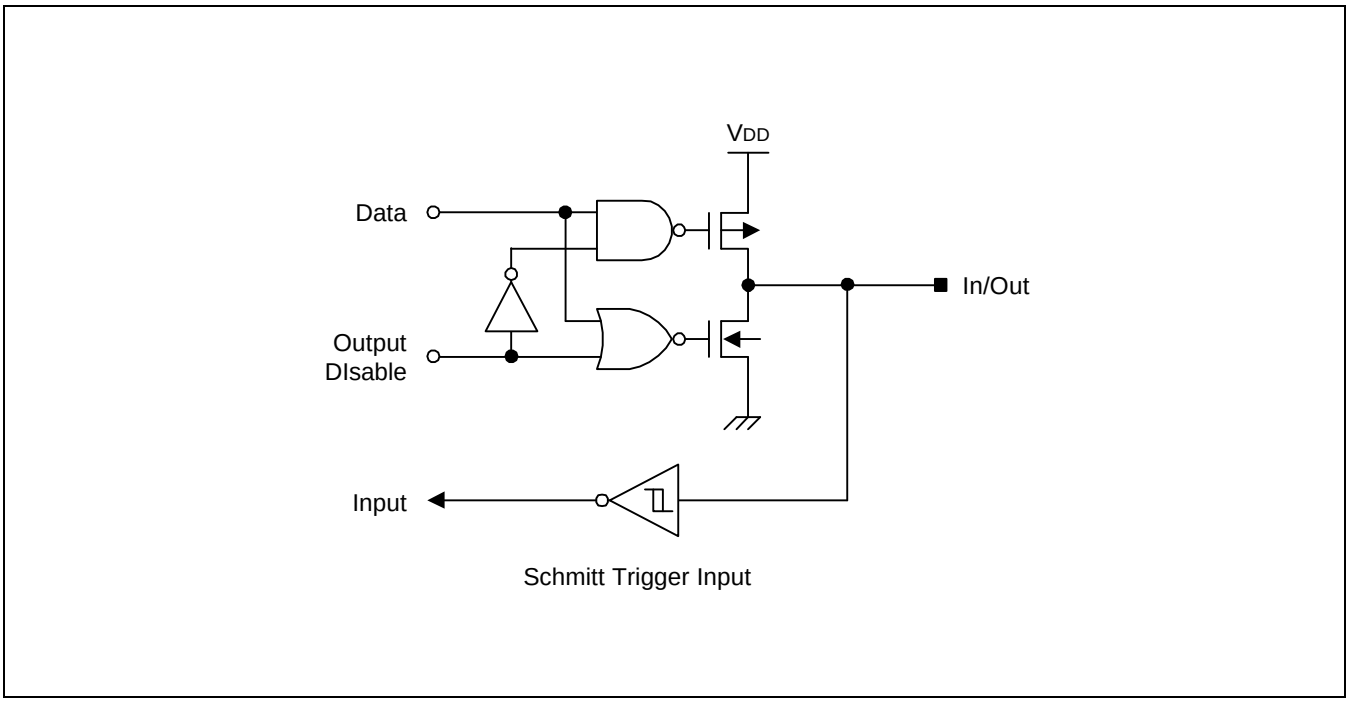


Figure 1-8. Pin Circuit Type 6 (P0.0, P1.0-P1.3, P2.4-P2.7, P3.0, OSDHT, PWM0)

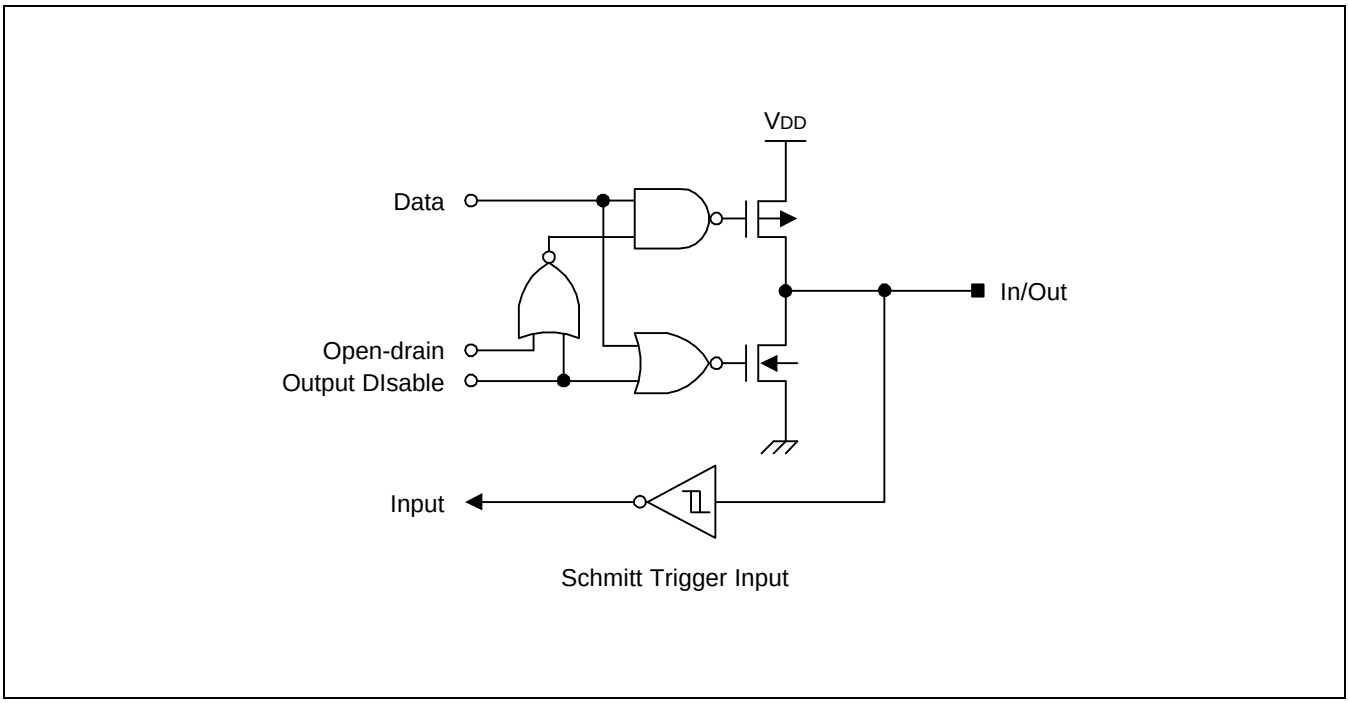


Figure 1-9. Pin Circuit type 7 (P0.4-P0.7)

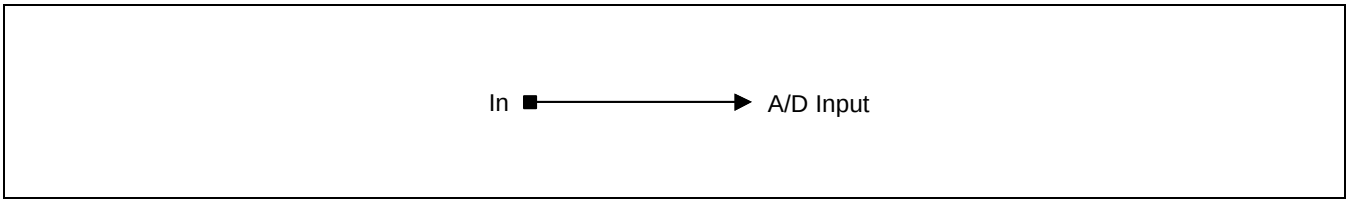


Figure 1-10. Pin Circuit type 8 (CVI IN, ADC0)

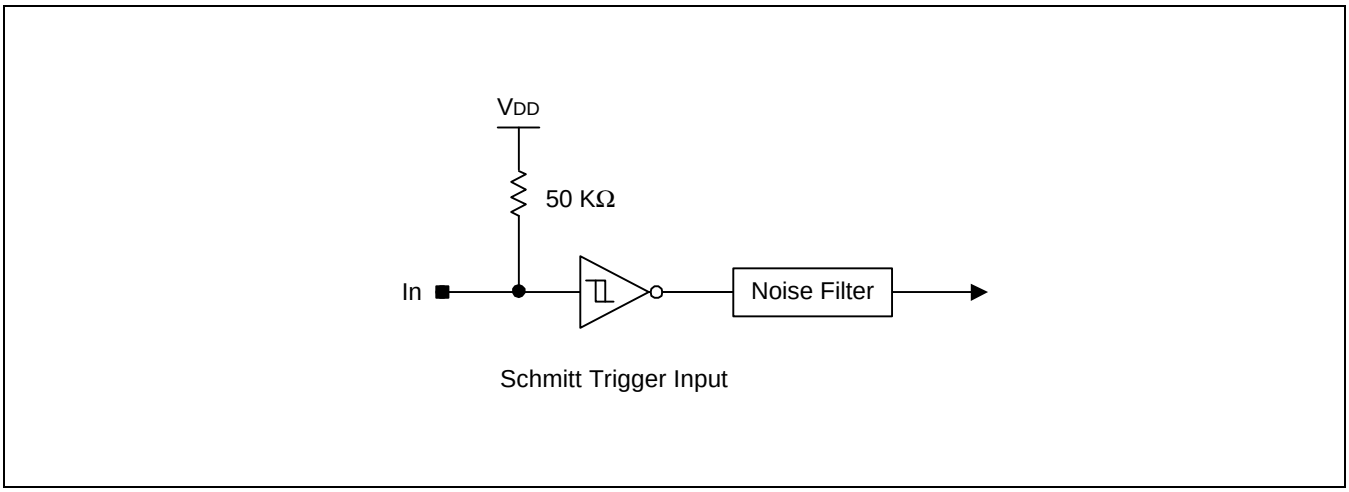


Figure 1-11. Pin Circuit type 9 (RESET)

CPU CORE OVERVIEW

The S3C380D CPU core is the ARM7TDMI processor, a general purpose, 32-bit microprocessor developed by Advanced RISC Machines, Ltd. (ARM). The core's architecture is based on Reduced Instruction Set Computer (RISC) principles. The RISC architecture makes the instruction set and its related decoding mechanisms simpler and more efficient than with microprogrammed Complex Instruction Set Computer (CISC) systems. The resulting benefit is high instruction throughput and impressive real-time interrupt response. Pipelining is also employed so that all components of the processing and memory systems can operate continuously. The ARM7TDMI has a 32-bit address bus.

An important feature of the ARM7TDMI processor, differentiating it from the ARM7 processor, is a unique architectural strategy called *THUMB*. The THUMB strategy is an extension of the basic ARM architecture and consists of 36 instruction formats. These formats are based on the standard 32-bit ARM instruction set, but have been re-coded using 16-bit wide opcodes.

Because THUMB instructions are one-half the bit width of normal ARM instructions, they produce very high-density code. When a THUMB instruction is executed, its 16-bit opcode is decoded by the processor into its equivalent instruction in the standard ARM instruction set. The ARM core then processes the 16-bit instruction as it would a normal 32-bit instruction. In other words, the Thumb architecture gives 16-bit systems a way to access the 32-bit performance of the ARM core without incurring the full overhead of 32-bit processing.

Because the ARM7TDMI core can execute both standard 32-bit ARM instructions and 16-bit Thumb instructions, it lets you mix routines of Thumb instructions and ARM code in the same address space. In this way, you can adjust code size and performance, routine by routine, to find the best programming solution for a specific application.

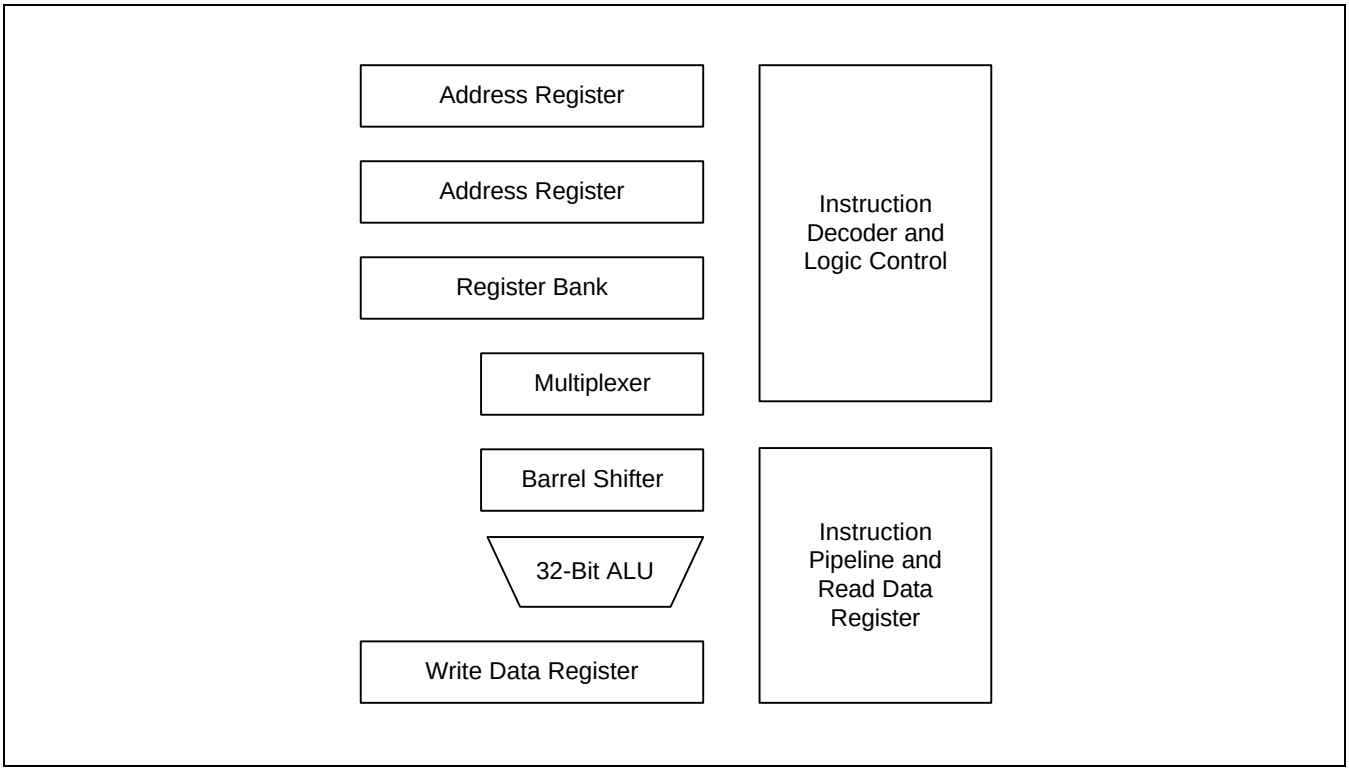


Figure 1-12. ARM7TDMI Core Block Diagram

INSTRUCTION SET

The S3C380D instruction set is divided into two subsets: a standard *32-bit ARM instruction set* and a *16-bit THUMB instruction set*.

The 32-bit ARM instruction set is comprised of thirteen basic instruction types which can, in turn, be divided into four broad classes:

- Four types of branch instructions which control program execution flow, instruction privilege levels, and switching between ARM code and THUMB code.
- Three types of data processing instructions which use the on-chip ALU, barrel shifter, and multiplier to perform high-speed data operations in a bank of 31 registers (all with 32-bit register widths).
- Three types of load and store instructions which control data transfer between memory locations and the registers. One type is optimized for flexible addressing, another for rapid context switching, and the third for swapping data.
- Three types of co-processor instructions which are dedicated to controlling external co-processors. These instructions extend the off-chip functionality of the instruction set in an open and uniform way.

NOTE

All 32-bit ARM instructions can be executed conditionally.

The 16-bit THUMB instruction set contains 36 instruction formats drawn from the standard 32-bit ARM instruction set. The THUMB instructions can be divided into four functional groups:

- Four branch instructions.
- Twelve data processing instructions, which are a subset of the standard ARM data processing instructions.
- Eight load and store register instructions.
- Four load and store multiple instructions.

NOTE

Each 16-bit THUMB instruction has a corresponding 32-bit ARM instruction with the identical processing model.

The 32-bit ARM instruction set and the 16-bit THUMB instruction sets are good targets for compilers of many different high-level languages. When assembly code is required for critical code segments, the ARM programming technique is straightforward, unlike that of some RISC processors which depend on sophisticated compiler technology to manage complicated instruction interdependencies.

Pipelining is employed so that all parts of the processor and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory.

OPERATING STATES

From a programmer's point of view, the ARM7TDMI core is always in one of two operating states. These states, which can be switched by software or by exception processing, are:

- *ARM state* (when executing 32-bit, word-aligned, ARM instructions), and
- *THUMB state* (when executing 16-bit, half-word aligned THUMB instructions).

OPERATING MODES

The ARM7TDMI core supports seven operating modes:

- *User mode*: the normal program execution state
- *FIQ (Fast Interrupt Request) mode*: for supporting a specific data transfer or channel process
- *IRQ (Interrupt ReQuest) mode*: for general purpose interrupt handling
- *Supervisor mode*: a protected mode for the operating system
- *Abort mode*: entered when a data or instruction pre-fetch is aborted
- *System mode*: a privileged user mode for the operating system
- *Undefined mode*: entered when an undefined instruction is executed

Operating mode changes can be controlled by software, or they can be caused by external interrupts or exception processing. Most application programs execute in User mode. Privileged modes (that is, all modes other than User mode) are entered to service interrupts or exceptions, or to access protected resources.

REGISTERS

The S3C380D CPU core has a total of 37 registers: 31 general-purpose, 32-bit registers, and 6 status registers. Not all of these registers are always available. Which registers are available to the programmer at any given time depends on the current processor operating state and mode.

NOTE

When the S3C380D is operating in ARM state, 16 general registers and one or two status registers can be accessed at any time. In privileged mode, mode-specific banked registers are switched in.

Two register sets, or banks, can also be accessed, depending on the core's current state: the *ARM state register set* and the *THUMB state register set*:

- The ARM state register set contains 16 directly accessible registers: R0-R15. All of these registers, except for R15, are for general-purpose use, and can hold either data or address values. An additional (seventeenth) register, the CPSR (Current Program Status Register), is used to store status information.
- The THUMB state register set is a subset of the ARM state set. You can access eight general registers, R0-R7, as well as the program counter (PC), a stack pointer register (SP), a link register (LR), and the CPSR. Each privileged mode has a corresponding banked stack pointer, link register, and saved process status register (SPSR).

The THUMB state registers are related to the ARM state registers as follows:

- THUMB state R0-R7 registers and ARM state R0-R7 registers are identical
- THUMB state CPSR and SPSRs and ARM state CPSR and SPSRs are identical
- THUMB state SP, LR, and PC map directly to ARM state registers R13, R14, and R15, respectively

In THUMB state, registers R8-R15 are not part of the standard register set. However, you can access them for assembly language programming and use them for fast temporary storage, if necessary.

EXCEPTIONS

An *exception* arises whenever the normal flow of program execution is interrupted. For example, when processing must be diverted to handle an interrupt from a peripheral. The processor's state just prior to handling the exception must be preserved so that the program flow can be resumed when the exception routine is completed. Multiple exceptions may arise simultaneously.

To process exceptions, the S3C380D uses the banked core registers to save the current state. The old PC value and the CPSR contents are copied into the appropriate R14 (LR) and SPSR register. The PC and mode bits in the CPSR are forced to a value which corresponds to the type of exception being processed.

The S3C380D core supports seven types of exceptions. Each exception has a fixed priority and a corresponding privileged processor mode, as shown in Table 1-2.

Table 1-2. S3C380D CPU Exceptions

Exception	Mode on Entry	Priority
Reset	Supervisor mode	1 (Highest)
Data abort	Abort mode	2
FIQ	FIQ mode	3
IRQ	IRQ mode	4
Prefetch abort	Abort mode	5
Undefined instruction	Undefined mode	6 (Lowest)
Software interrupt	Supervisor mode	6 (Lowest)

2 PROGRAMMER'S MODEL

OVERVIEW

The S3C380D was developed using the advanced ARM7TDMI core designed by Advanced RISC Machines, Ltd.

PROCESSOR OPERATING STATES

From the programmer's point of view, the ARM7TDMI can be in one of the following two states:

- ARM state which executes 32-bit, word-aligned ARM instructions.
- *THUMB state* which operates with 16-bit, halfword-aligned THUMB instructions. In this state, the PC uses bit 1 to select among alternate halfwords.

NOTE

Transition between these two states does not affect the processor mode or the contents of the registers.

SWITCHING STATE

Entering THUMB State

Entry into THUMB state can be achieved by executing a BX instruction with the state bit (bit 0) set in the operand register.

Transition to THUMB state will also occur automatically on return from an exception (IRQ, FIQ, UNDEF, ABORT, SWI etc.), if the exception was entered with the processor in THUMB state.

Entering ARM State

Entry into ARM state happens:

- On execution of the BX instruction with the state bit clear in the operand register.
- On the processor taking an exception (IRQ, FIQ, RESET, UNDEF, ABORT, SWI etc.). In this case, the PC is placed in the exception mode's link register, and execution commences at the exception's vector address.

MEMORY FORMATS

ARM7TDMI views memory as a linear collection of bytes numbered upwards from zero. Bytes 0 to 3 hold the first word stored, bytes 4 to 7 the second, and so on. ARM7TDMI can treat words in memory as being stored either in Big-Endian or Little-Endian format.

NOTE

The S3C380D is configured to Little-Endian format.

BIG-ENDIAN FORMAT

In Big-Endian format, the most significant byte of a word is stored at the lowest numbered byte and the least significant byte at the highest numbered byte. Byte 0 of the memory system is therefore connected to data lines 31 through 24.

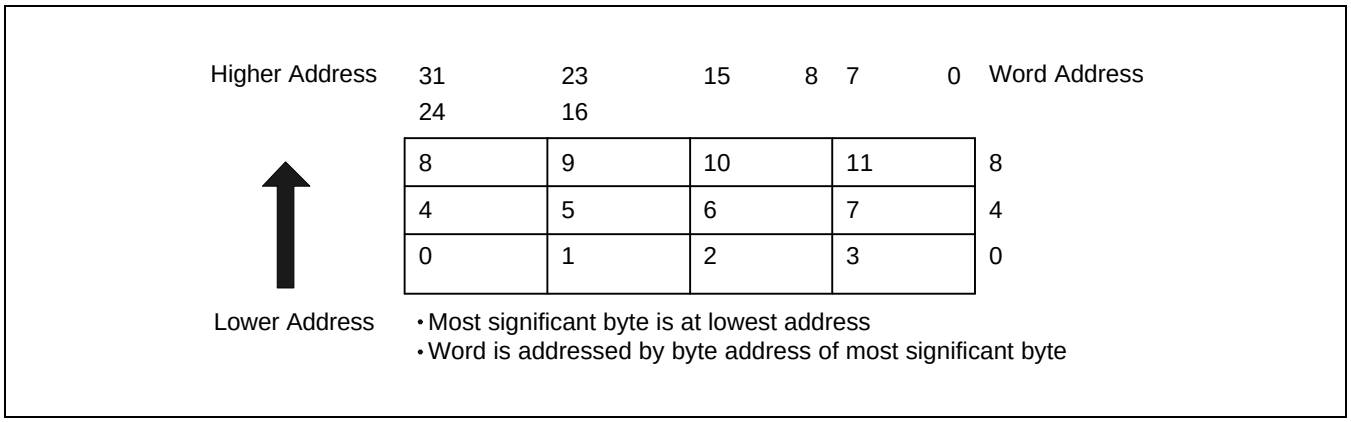


Figure 2-1. Big-Endian Addresses of Bytes within Words

LITTLE-ENDIAN FORMAT

In Little-Endian format, the lowest numbered byte in a word is considered the word's least significant byte, and the highest numbered byte the most significant. Byte 0 of the memory system is therefore connected to data lines 7 through 0.

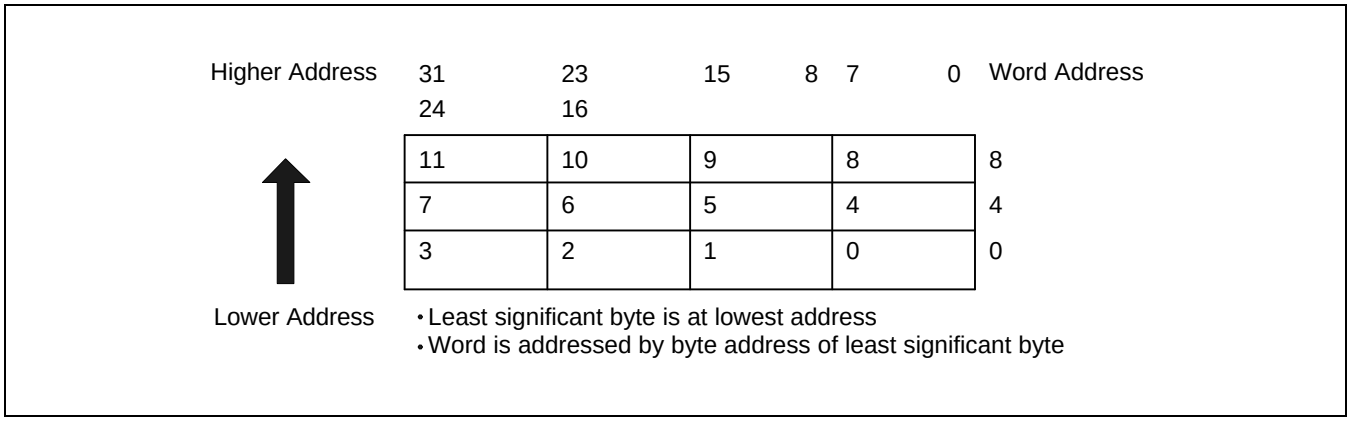


Figure 2-2. Little-Endian Addresses of Bytes whthin Words

INSTRUCTION LENGTH

Instructions are either 32 bits (in ARM state) or 16 bits long (in THUMB state).

Data Types

ARM7TDMI supports byte (8-bit), halfword (16-bit), and word (32-bit) data types. Words must be aligned to four-byte boundaries and half words to two-byte boundaries.

OPERATING MODES

ARM7TDMI supports seven modes of operation:

- User (usr): The normal ARM program execution state
- FIQ (fiq): Designed to support a data transfer or channel process
- IRQ (irq): Used for general-purpose interrupt handling
- Supervisor (svc): Protected mode for the operating system
- Abort mode (abt): Entered after a data or instruction prefetch abort
- System (sys): A privileged user mode for the operating system
- Undefined (und): Entered when an undefined instruction is executed

Mode changes may be made under software control, or may be brought about by external interrupts or exception processing. Most application programs execute in User mode. The non-user modes-known as privileged modes-are entered in order to service interrupts or exceptions, or to access protected resources.

REGISTERS

ARM7TDMI has a total of 37 registers - 31 general-purpose 32-bit registers and six status registers - but all these cannot be seen at once. The processor state and operating mode dictate which registers are available to the programmer.

The ARM State Register Set

In ARM state, 16 general registers and one or two status registers are visible at any one time. In privileged (non-User) modes, mode-specific banked registers are switched in. Figure 2-3 shows which registers are available in each mode: the banked registers are marked with a shaded triangle.

The ARM state register set contains 16 directly accessible registers: R0 to R15. All of these except R15 are general-purpose, and may be used to hold either data or address values. In addition to these, there is a seventeenth register used to store status information.

Register 14	is used as the subroutine link register. This receives a copy of R15 when a Branch and Link (BL) instruction is executed. At all other times it may be treated as a general-purpose register. The corresponding banked registers R14_svc, R14_irq, R14_fiq, R14_abt and R14_und are similarly used to hold the return values of R15 when interrupts and exceptions arise, or when Branch and Link instructions are executed within interrupt or exception routines.
Register 15	holds the Program Counter (PC). In ARM state, bits [1:0] of R15 are zero and bits [31:2] contain the PC. In THUMB state, bit [0] is zero and bits [31:1] contain the PC.
Register 16	is the CPSR (Current Program Status Register). This contains condition code flags and the current mode bits.

FIQ mode has seven banked registers mapped to R8-14 (R8_fiq-R14_fiq). In ARM state, many FIQ handlers do not need to save any register. User, IRQ, Supervisor, Abort and Undefined each have two banked registers mapped to R13 and R14, allowing each of these modes to have a private stack pointer and link registers.

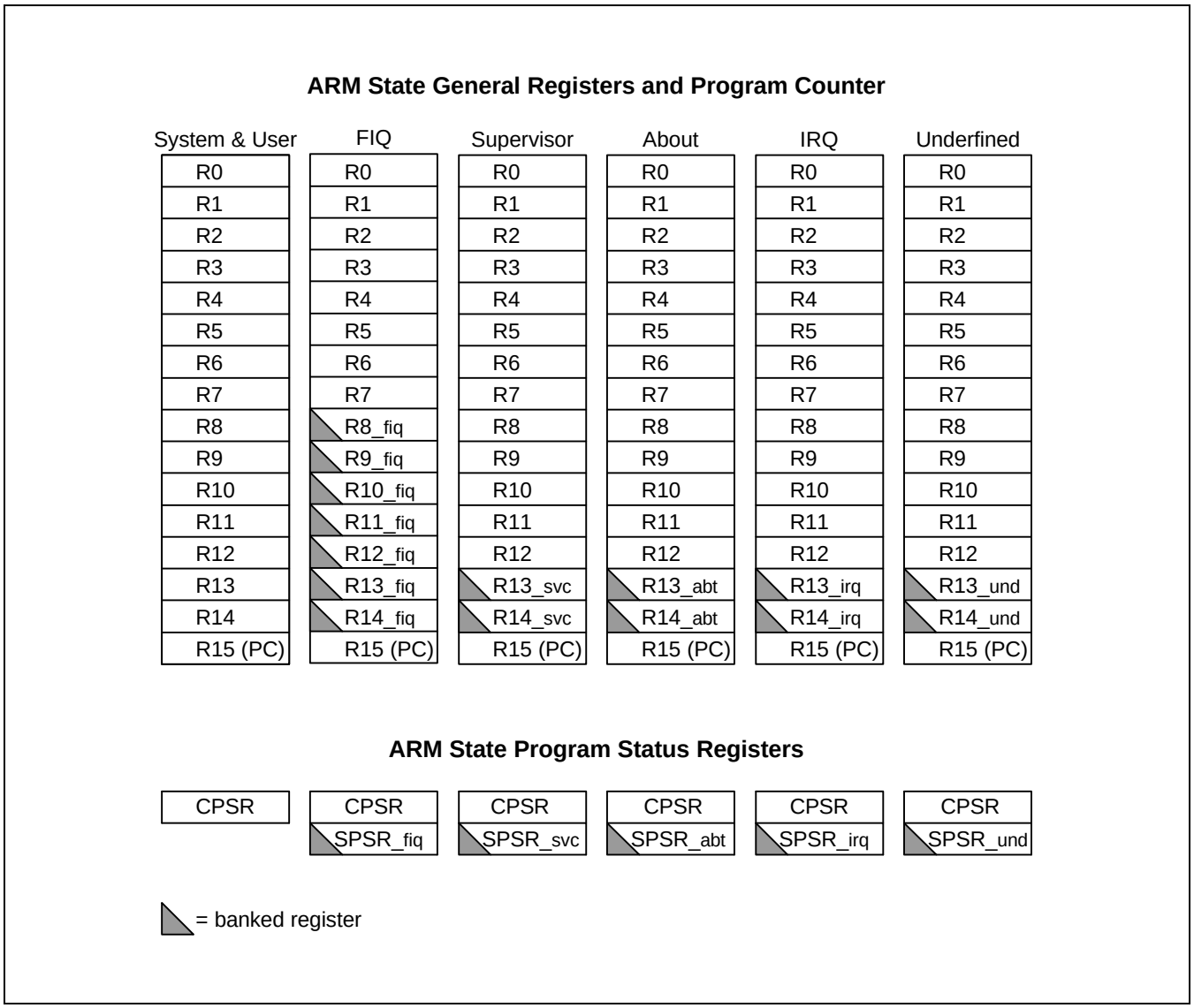


Figure 2-3. Register Organization in ARM State

The THUMB State Register Set

The THUMB state register set is a subset of the ARM state set. The programmer has direct access to eight general registers, R0-R7, as well as the Program Counter (PC), a stack pointer register (SP), a link register (LR), and the CPSR. There are banked Stack Pointers, Link Registers and Saved Process Status Registers (SPSRs) for each privileged mode. This is shown in Figure 2-4.

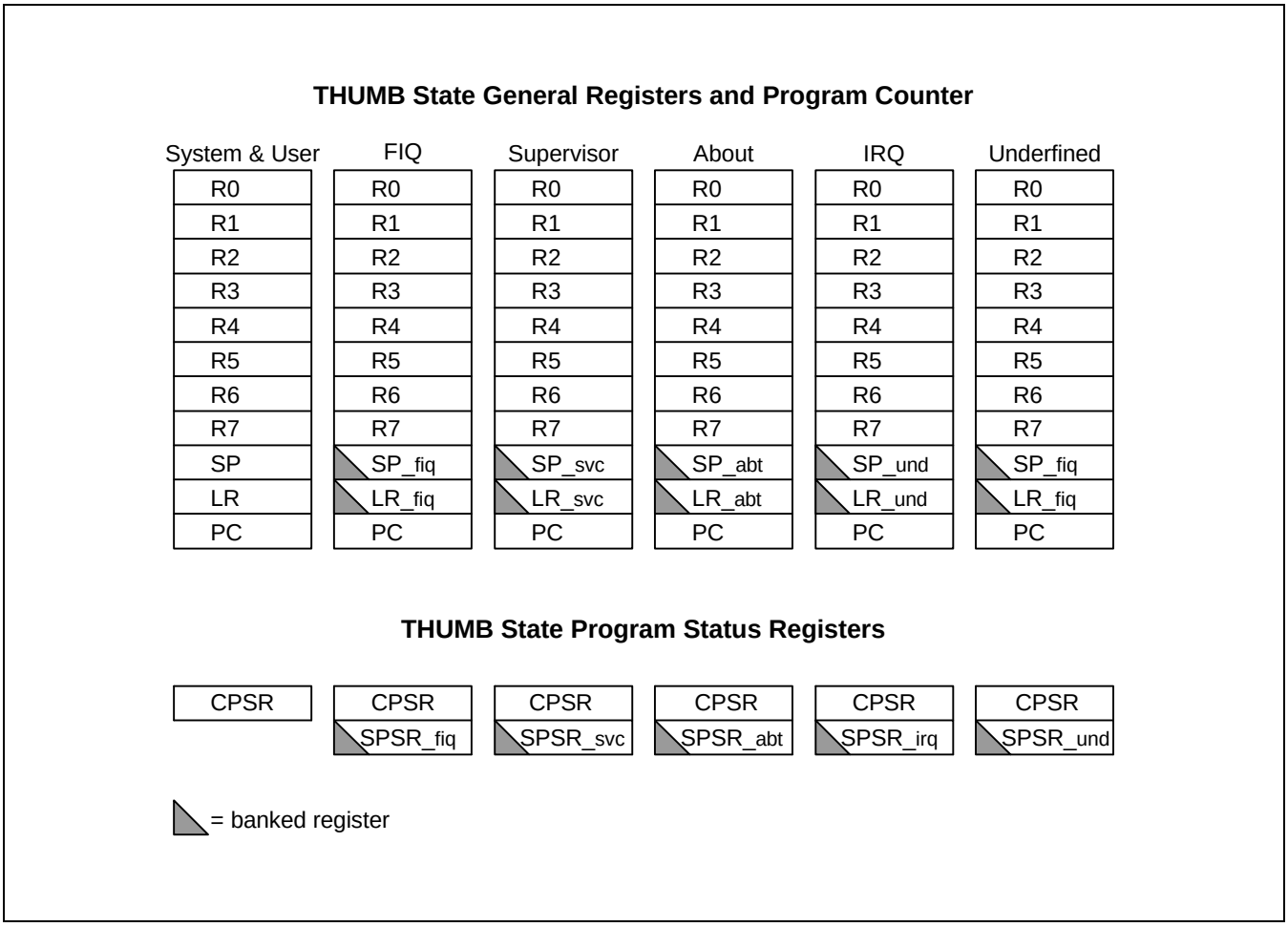


Figure 2-4. Register Organization in THUMB state

The relationship between ARM and THUMB state registers

The THUMB state registers relate to the ARM state registers in the following way:

- THUMB state R0-R7 and ARM state R0-R7 are identical
- THUMB state CPSR and SPSRs and ARM state CPSR and SPSRs are identical
- THUMB state SP maps onto ARM state R13
- THUMB state LR maps onto ARM state R14
- The THUMB state Program Counter maps onto the ARM state Program Counter (R15)

This relationship is shown in Figure 2-5.

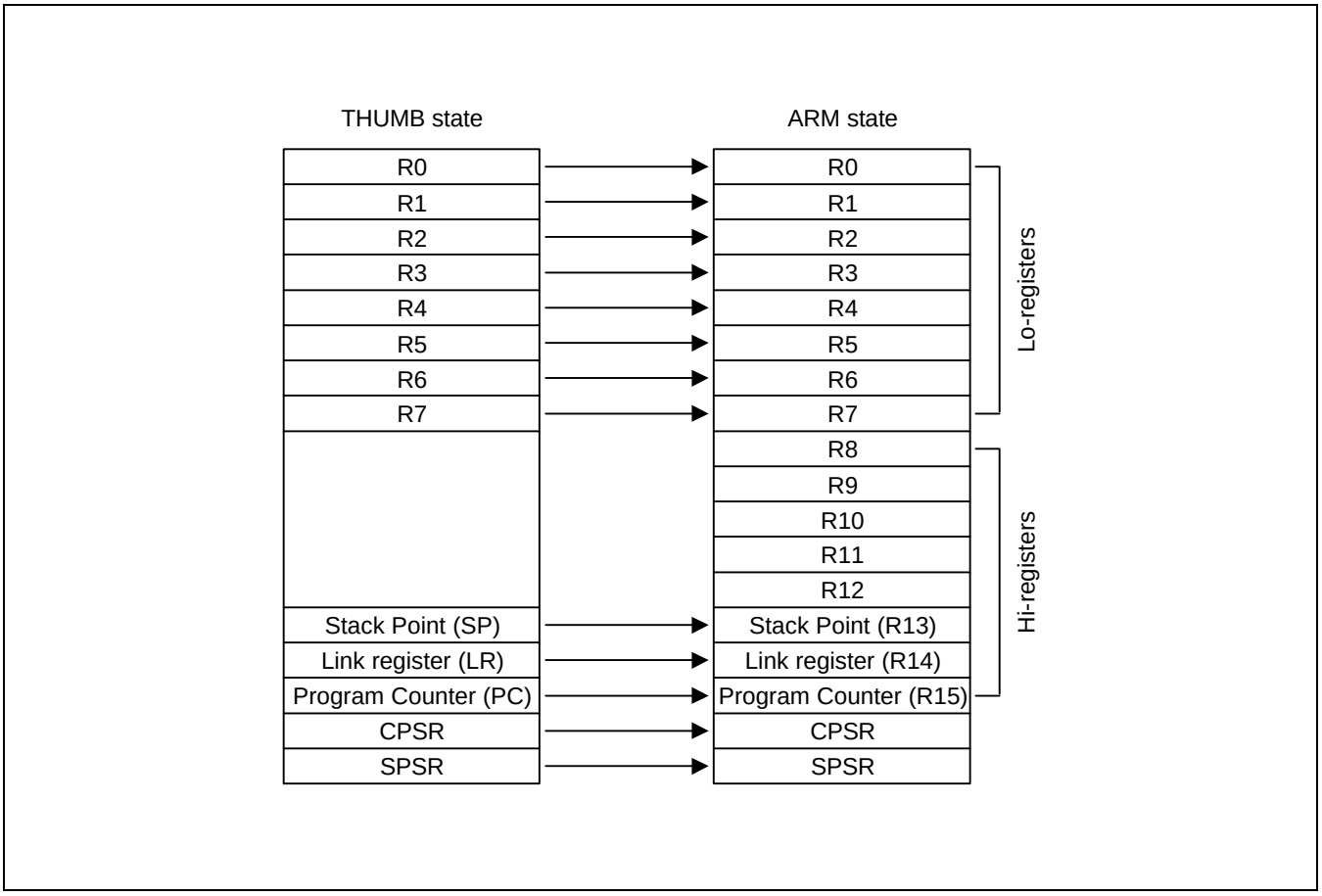


Figure 2-5. Mapping of THUMB State Registers onto ARM State Registers

Accessing Hi-Registers in THUMB State

In THUMB state, registers R8-R15 (the Hi registers) are not part of the standard register set. However, the assembly language programmer has limited access to them, and can use them for fast temporary storage.

A value may be transferred from a register in the range R0-R7 (a Lo register) to a Hi register, and from a Hi register to a Lo register, using special variants of the MOV instruction. Hi register values can also be compared with or added to Lo register values with the CMP and ADD instructions. For more information, refer to Figure 3-34.

THE PROGRAM STATUS REGISTERS

The ARM7TDMI contains a Current Program Status Register (CPSR), plus five Saved Program Status Registers (SPSRs) for use by exception handlers. Their functions are as follows:

- Hold information about the most recently performed ALU operation
- Control enabling and disabling of interrupts
- Set the processor operating mode

The arrangement of bits is shown in Figure 2-6.

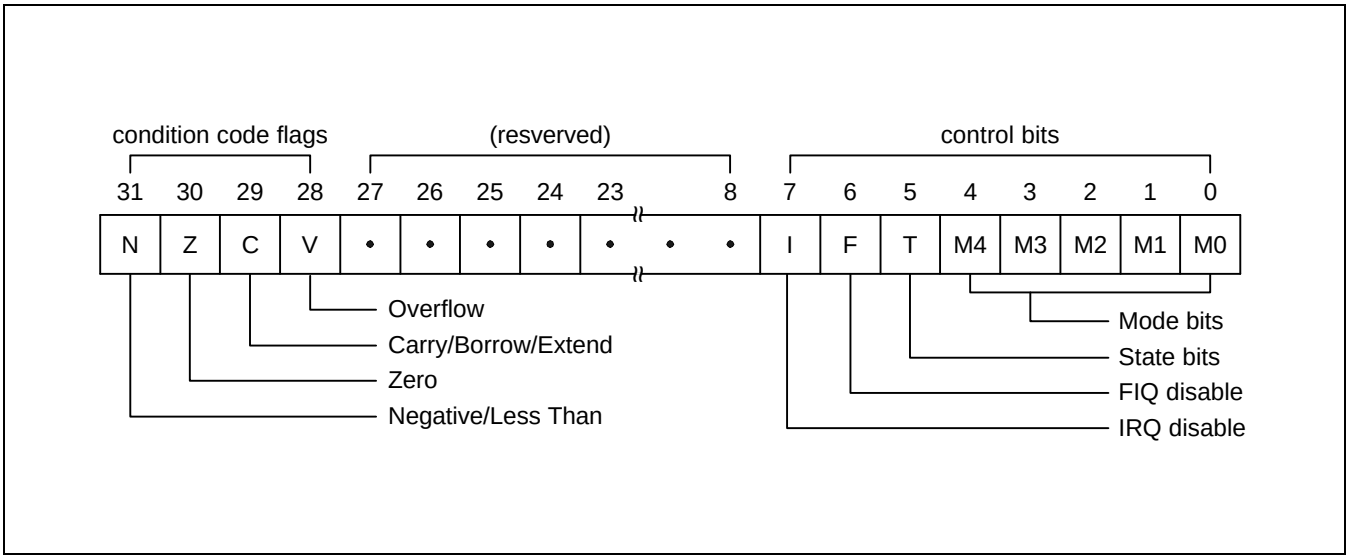


Figure 2-6. Program Status Register Format

The Condition Code Flags

The N, Z, C and V bits are the condition code flags. These may be changed as a result of arithmetic and logical operations, and may be tested to determine whether an instruction should be executed.

In ARM state, all instructions may be executed conditionally: see Table 3-2 for details.

In THUMB state, only the Branch instruction is capable of conditional execution. See Figure 3-46 for details.

The Control Bits

The bottom 8 bits of a PSR (incorporating I, F, T and M[4:0]) are known collectively as the control bits. These will change when an exception arises. If the processor is operating in a privileged mode, they can also be manipulated by software.

The T bit — This reflects the operating state. When this bit is set, the processor is executing in THUMB state, otherwise it is executing in ARM state. This is reflected on the **TBIT** external signal.

Note that the software must never change the state of the **TBIT** in the CPSR. Otherwise, the processor will enter an unpredictable state.

Interrupt disable bits — The I and F bits are the interrupt disable bits. When set, they disable the IRQ and FIQ interrupts respectively.

The mode bits — The M4, M3, M2, M1, and M0 bits (M[4:0]) are the mode bits. They determine the processor's operating mode, as shown in Table 2-1. Not all combinations of the mode bits define a valid processor mode. Only those explicitly described shall be used. The user should be aware that if any illegal value is programmed into the mode bits, M[4:0], the processor will enter an unrecoverable state. If this occurs, a reset should be applied.

Table 2-1. PSR Mode Bit Values

M[4:0]	Mode	Visible THUMB state registers	Visible ARM atate registers
10000	User	R7..R0, LR, SP PC, CPSR	R14..R0, PC, CPSR
10001	FIQ	R7..R0, LR_fiq, SP_fiq PC, CPSR, SPSR_fiq	R7..R0, R14_fiq..R8_fiq, PC, CPSR, SPSR_fiq
10010	IRQ	R7..R0, LR_irq, SP_irq PC, CPSR, SPSR_irq	R12..R0, 14_irq..R13_irq, PC, CPSR, SPSR_irq
10011	Supervisor	R7..R0, LR_svc, SP_svc, PC, CPSR, SPSR_svc	R12..R0, R14_svc..R13_svc, PC, CPSR, SPSR_svc
10111	Abort	R7..R0, LR_abt, SP_abt, PC, CPSR, SPSR_abt	R12..R0, R14_abt..R13_abt, PC, CPSR, SPSR_abt
11011	Undefined	R7..R0 LR_und, SP_und, PC, CPSR, SPSR_und	R12..R0, R14_und..R13_und, PC, CPSR
11111	System	R7..R0, LR, SP PC, CPSR	R14..R0, PC, CPSR

Reserved bits — The remaining bits in the PSRs are reserved. When changing a PSR's flag or control bits, you must ensure that these unused bits are not altered. Also, your program should not rely on them containing specific values, since in future processors they may read as one or zero.

EXCEPTIONS

Exceptions arise whenever the normal flow of a program has to be halted temporarily, for example to service an interrupt from a peripheral. Before an exception can be handled, the current processor state must be maintained so that the original program can resume when the handler routine finishes.

It is possible for several exceptions to arise at the same time. If this happens, they are dealt with in a fixed order. See Exception Priorities on page 2-14.

Action on Entering an Exception

When handling an exception, the ARM7TDMI would do the following:

1. Contains the address of the next instruction in the appropriate Link Register. If the exception has been entered from ARM state, the address of the next instruction is copied into the Link Register (that is, current PC + 4 or PC + 8 depending on the exception. See Table 2-2 on for details). If the exception has been entered from THUMB state, the value written into the Link Register is the current PC offset by a value such that the program resumes from the correct place on return from the exception. This means that the exception handler does not need to determine which state the exception was entered from. For example, in the case of SWI, MOVS PC, R14_svc will always return to the next instruction regardless of whether the SWI was executed in ARM or THUMB state.
2. Copies the CPSR into the appropriate SPSR
3. Forces the CPSR mode bits to a value which depends on the exception
4. Forces the PC to fetch the next instruction from the relevant exception vector

It may also set the interrupt disable flags to prevent otherwise unmanageable nestings of exceptions.

If the processor was in THUMB state when an exception occured, it would automatically switch into ARM state when the PC is loaded with the exception vector address.

Action on Leaving an Exception

On completion, the exception handler would do the following:

1. Moves the Link Register, minus an offset where appropriate, to the PC. (The offset will vary depending on the type of exception.)
2. Copies the SPSR back to the CPSR
3. Clears the interrupt disable flags, if they were set on entry

NOTE

An explicit switch back to THUMB state is never needed, since restoring the CPSR from the SPSR automatically sets the T bit to the value it held immediately prior to the exception.

Exception Entry/Exit Summary

Table 2-2 summarises the PC value contained in the relevant R14 on exception entry, and the recommended instruction for exiting the exception handler.

Table 2-2. Exception Entry/Exit

	Return Instruction	Previous State		Notes
		ARM R14_x	THUMB R14_x	
BL	MOV PC, R14	PC + 4	PC + 2	1
SWI	MOVS PC, R14_svc	PC + 4	PC + 2	1
UDEF	MOVS PC, R14_und	PC + 4	PC + 2	1
FIQ	SUBS PC, R14_fiq, #4	PC + 4	PC + 4	2
IRQ	SUBS PC, R14_irq, #4	PC + 4	PC + 4	2
PABT	SUBS PC, R14_abt, #4	PC + 4	PC + 4	1
DABT	SUBS PC, R14_abt, #8	PC + 8	PC + 8	3
RESET	NA	–	–	4

NOTES:
1. Where PC is the address of the BL/SWI/Undefined Instruction fetch which had the prefetch abort.
2. Where PC is the address of the instruction which was not executed since the FIQ or IRQ took priority.
3. Where PC is the address of the Load or Store instruction which generated the data abort.
4. The value saved in R14_svc upon reset is unpredictable.

FIQ

The FIQ (Fast Interrupt Request) exception is designed to support a data transfer or channel process, and in ARM state, it has sufficient private registers to remove the need for register saving (thus minimising the overhead of context switching).

FIQ is externally generated by taking the **nFIQ** input LOW. This input can except either synchronous or asynchronous transitions, depending on the state of the **ISYNC** input signal. When **ISYNC** is LOW, **nFIQ** and **nIRQ** are considered asynchronous, and a cycle delay for synchronization is incurred before the interrupt can affect the processor flow.

Irrespective of whether the exception was entered from ARM or Thumb state, a FIQ handler should leave the interrupt by executing

```
SUBS    PC,R14_fiq,#4
```

FIQ may be disabled by setting the CPSR's F flag (but note that this is not possible in User mode). If the F flag is clear, ARM7TDMI checks for a LOW level on the output of the FIQ synchroniser at the end of each instruction.

IRQ

The IRQ (Interrupt Request) exception is a normal interrupt caused by a LOW level on the **nIRQ** input. IRQ has a lower priority than FIQ and is masked out when an FIQ sequence is entered. It may be disabled at any time by setting the I bit in the CPSR, though this can only be done in a privileged (non-User) mode.

Irrespective of whether the exception was entered from ARM or Thumb state, an IRQ handler should return from the interrupt by executing

```
SUBS    PC,R14_irq,#4
```

Abort

An abort indicates that the current memory access cannot be completed. It can be signalled by the external **ABORT** input. ARM7TDMI checks for the abort exception during memory access cycles.

There are two types of abort:

- *Prefetch abort:* occurs during an instruction prefetch.
- *Data abort:* occurs during a data access.

If a prefetch abort occurs, the prefetched instruction is marked as invalid, but the exception will not be taken until the instruction reaches the head of the pipeline. If the instruction is not executed - for example because a branch occurs while it is in the pipeline - the abort does not take place.

If a data abort occurs, the action taken depends on the instruction type:

- Single data transfer instructions (LDR, STR) write back modified base registers: the abort handler must be aware of this.
- The swap instruction (SWP) is aborted as though it has not been executed.
- Block data transfer instructions (LDM, STM) complete. If write-back is set, the base is updated. If the instruction would have overwritten the base with data (i.e. it has the base in the transfer list), the overwriting is prevented. Any register overwriting is prevented after an abort is indicated, which means in particular that R15 (always the last register to be transferred) is preserved in an aborted LDM instruction.

The abort mechanism allows the implementation of a demand paged virtual memory system. In such a system the processor is allowed to generate arbitrary addresses. When the data at an address is unavailable, the Memory Management Unit (MMU) signals an abort. The abort handler must then figure out the cause of the abort, make the requested data available, and retry the aborted instruction. The application program needs no knowledge of the amount of memory available to it, nor is its state in any way affected by the abort.

After resolving the cause of the abort, the handler should execute the following, irrespective of the state (ARM or Thumb):

```
SUBS    PC,R14_abt,#4    ;  for a prefetch abort, or
SUBS    PC,R14_abt,#8    ;  for a data abort
```

This restores both the PC and the CPSR, and retries the aborted instruction.

Software Interrupt

The software interrupt instruction (SWI) is used for entering Supervisor mode, usually to request a particular supervisor function. A SWI handler should return by executing the following, irrespective of the state (ARM or Thumb):

```
MOV     PC,R14_svc
```

This restores the PC and the CPSR, and returns to the instruction following the SWI.

NOTE

nFIQ, nIRQ, ISYNC, LOCK, BIGEND, and ABORT pins exist only in the ARM7TDMI CPU core.

Undefined Instruction

When ARM7TDMI comes across an instruction which it cannot handle, it takes the undefined instruction trap. This mechanism may be used to extend either the THUMB or ARM instruction set by software emulation.

After emulating the failed instruction, the trap handler should execute the following, irrespective of the state (ARM or Thumb):

```
MOVS    PC,R14_und
```

This restores the CPSR and returns to the instruction following the undefined instruction.

Exception Vectors

The following table shows the exception vector addresses.

Table 2-3. Exception Vectors

Address	Exception	Mode in Entry
0x00000000	Reset	Supervisor
0x00000004	Undefined instruction	Undefined
0x00000008	Software Interrupt	Supervisor
0x0000000C	Abort (prefetch)	Abort
0x00000010	Abort (data)	Abort
0x00000014	Reserved	Reserved
0x00000018	IRQ	IRQ
0x0000001C	FIQ	FIQ

Exception Priorites

When multiple exceptions arise at the same time, a fixed priority system determines the order in which they are handled:

Highest priority:

1. Reset
2. Data abort
3. FIQ
4. IRQ
5. Prefetch abort

Lowest priority:

6. Undefined Instruction, Software interrupt.

Not All Exceptions Can Occur at Once:

Undefined Instruction and Software Interrupt are mutually exclusive, since they each correspond to particular (non-overlapping) decodings of the current instruction.

If a data abort occurs at the same time as a FIQ, and FIQs are enabled (i.e. the CPSR's F flag is clear), ARM7TDMI enters the data abort handler and then immediately proceeds to the FIQ vector. A normal return from FIQ will cause the data abort handler to resume execution. Placing data abort at a higher priority than FIQ is necessary to ensure that the transfer error does not avoid detection. The time for this exception entry should be added to worst-case FIQ latency calculations.

INTERRUPT LATENCIES

The worst case latency for FIQ, assuming that it is enabled, consists of the longest time the request can take to pass through the synchroniser (*Tsyncmax* if asynchronous), the time for the longest instruction to complete (*Tldm*, the longest instruction is an LDM which loads all the registers including the PC), the time for the data abort entry (*Texc*), plus the time for FIQ entry (*Tfiq*). At the end of this time, ARM7TDMI executes the instruction at 0x1C.

Tsyncmax is 3 processor cycles, *Tldm* is 20 cycles, *Texc* is 3 cycles, and *Tfiq* is 2 cycles. The total time is therefore 28 processor cycles. This is just over 1.4 microseconds in a system which uses a continuous 20 MHz processor clock. The maximum IRQ latency calculation is similar to this, but must allow for the fact that an FIQ has higher priority and could delay entry into the IRQ handling routine for an arbitrary length of time. The minimum latency for an FIQ or IRQ consists of the shortest time the request can take through the synchroniser (*Tsyncmin*) plus *Tfiq*. This is 4 processor cycles.

RESET

When the RESET signal goes LOW, ARM7TDMI abandons the executing instruction and continues to fetch instructions from incrementing word addresses.

When RESET goes HIGH again, ARM7TDMI would do the following:

1. Overwrites R14_svc and SPSR_svc by copying the current values of the PC and CPSR into them. The value of the saved PC and SPSR is not defined.
2. Forces M[4:0] to 10011 (Supervisor mode), sets the I and F bits in the CPSR, and clears the CPSR's T bit.
3. Forces the PC to fetch the next instruction from address 0x00.
4. Execution resumes in ARM state.

4 ADDRESS SPACES

OVERVIEW

The S3C380D microcontroller has two kinds of address space:

— Internal program memory (ROM)

— Internal register file

The S3C380D has an on-chip 128-Kbyte mask-programmable ROM. An external memory interface is not implemented.

There are 1504 general-purpose 16-bit data registers in the register file. Forty two 16-bit registers are used for CPU, interrupt and system control. To support peripheral, I/O, and clock functions, there are 27 control registers and 9 data registers. In addition, there is a 544 × 16-bit area for on-screen display (OSD) video RAM.

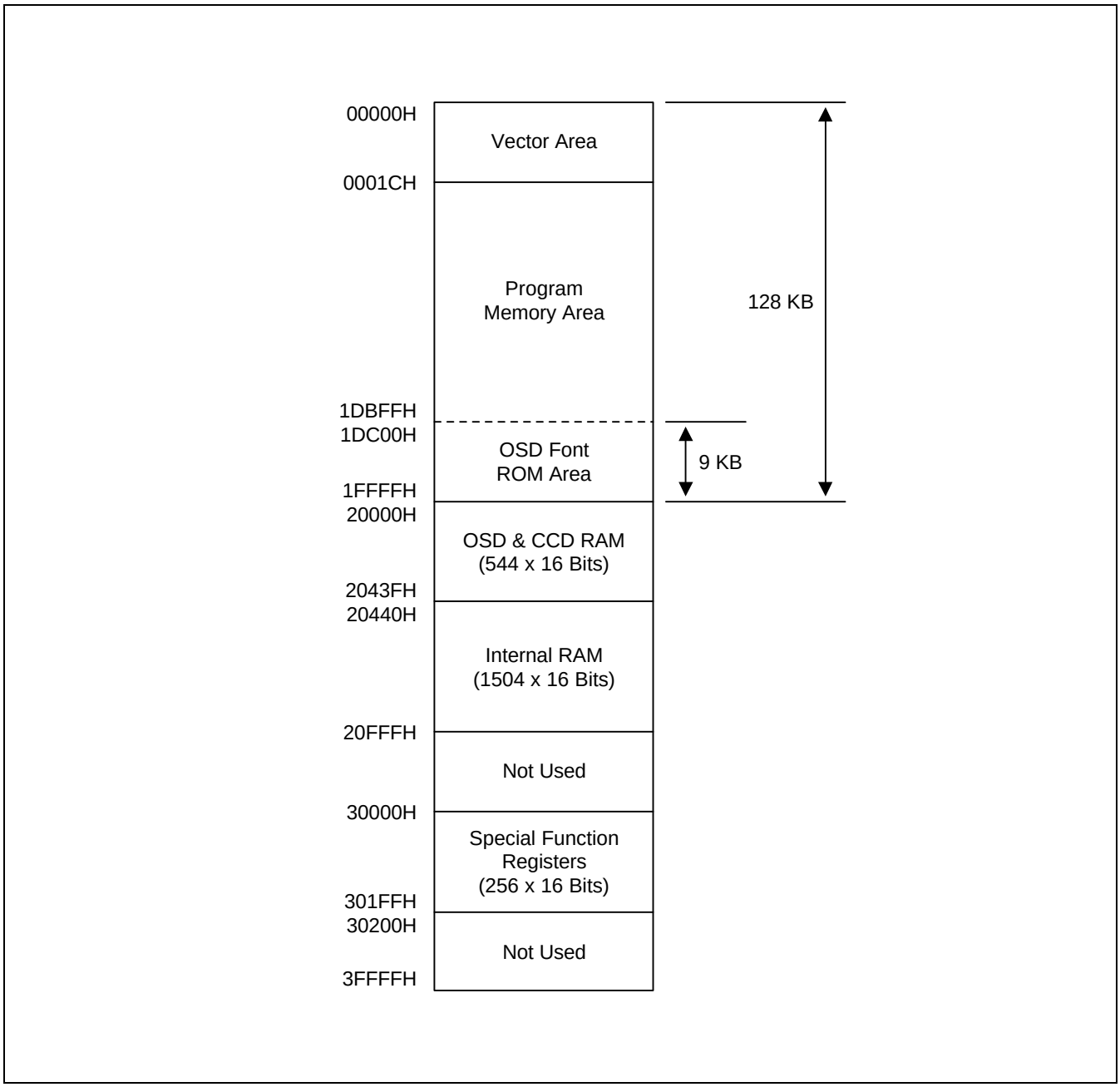


Figure 4-1. S3C380D/F380D Memory Map

PROGRAM MEMORY (ROM)

Program memory (ROM) stores program codes or table data. The S3C380D has a 128-Kbyte mask programmable program memory (1FFFFh).

As shown in Figure 2-1, the first 32 bytes of the ROM (0H-01FH) are reserved for interrupt vector addresses. Unused locations in this range can be used as normal program memory. When the vector address area is used to store normal program data, care must be taken to avoid overwriting vector addresses stored in these locations.

When you use S3C380D, the program memory is 128-Kbyte in total, including the 9-Kbyte OSD font ROM. If you use 9-Kbyte as the OSD font ROM area, you are able to use the remaining 119-Kbyte for program memory.

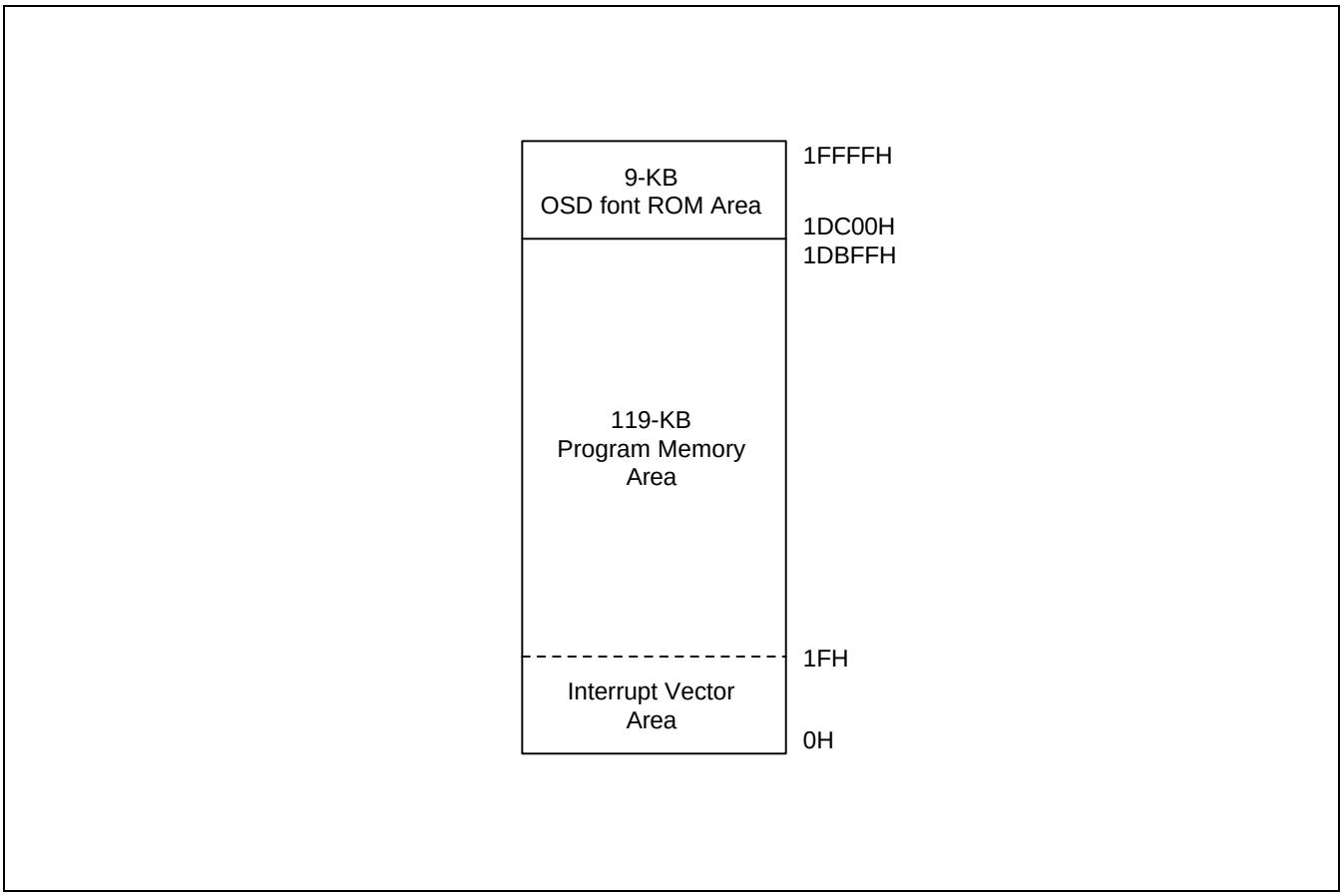


Figure 4-2. Program Memory Address Spaces

REGISTER ARCHITECTURE

The special function register has 78 halfwords, and the address exists within 30000H-301FFH. The OSD & CCD RAM have 544 halfwords, and the address exists within 20000H-2043FH. The general-purpose RAM has 1504 halfwords, and the address exists within 20440-20FFFH.

Table 4-1. Register Type Summary

Register Type	Number of Bytes	Number of Half-word
General-purpose registers	3008	1504
Special function registers	156	78
On-screen display (OSD) video RAM	1088	544
Total Addressable Bytes	4152	2126

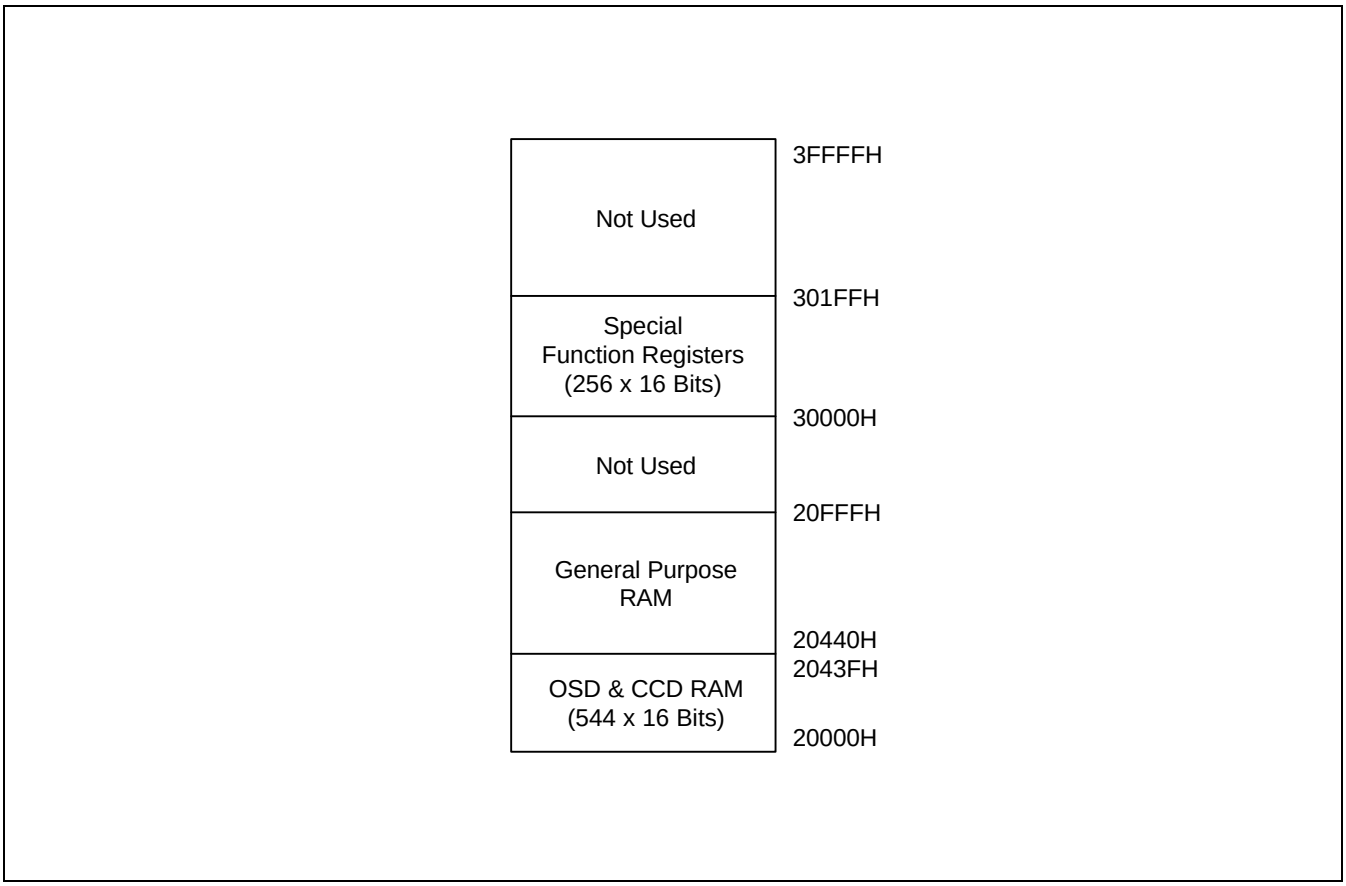


Figure 4-3. Internal Register File Organization

5 SPECIAL FUNCTION REGISTERS

OVERVIEW

In this chapter, detailed descriptions of the S3C380D/F380D control registers are presented in an easy-to-read format. These descriptions will help familiarize you with the mapped locations in the register file. You can also use them as a quick-reference source when writing application programs.

System and peripheral registers are summarized in Tables 5-1. Figure 5-1 illustrates the important features of the standard register description format.

Control register descriptions are arranged in alphabetical order according to register mnemonic. More information about control registers is presented in the context of the various peripheral hardware descriptions in this manual.

S3C380D SPECIAL REGISTERS

Table 5-1. Special Registers

Group	Registers	Address	Description	R/W	Reset Value
Basic and 16-bit timers	BTCON	0x30000	Watchdog timer control register	R/W	0000h
	BTCNT	0x30002	Basic timer counter register	R	xx00h
	T0DATA	0x30004	Timer 0 data register	R/W	0FFFFh
	T1DATA	0x30006	Timer 1 data register	R/W	0FFFFh
	T2DATA	0x30008	Timer 2 data register	R/W	0FFFFh
	TMCON	0x3000A	Timer control register	R/W	0000h
	T0CNT	0x3000C	Timer 0 counter register	R	xxxxh
	T1CNT	0x3000E	Timer 1 counter register	R	xxxxh
	T2CNT	0x30010	Timer 2 counter register	R	xxxxh
PWM	PWMCON	0x30012	PWM control register	R/W	xxx0h
	PWM0	0x30014	PWM 0 data register (14-bit)	R/W	3FC0h
	PWM1	0x30016	PWM 1 data register (14-bit)	R/W	3FC0h
Remocon receive	RRCR	0x30018	Remocon receive control register	R/W	0100h
	FIFOD	0x3001A	FIFO data register	R	xxxxh
Real time	RTCON	0x3001C	Real time clock control register	R/W	0000h
PLL	PLLCON	0x3001E	PLL clock control register	R/W	B763h
System	SYSCON	0x30020	System control register	R/W	0000h
Interrupt	INTPR	0x30022	Interrupt pending register	R/W	0000h
	INTMD	0x30024	Interrupt mode register	R/W	0000h
	INTMK	0x30026	Interrupt mask register	R/W	0000h
	IPR0	0x30028	Interrupt priority register 0	R/W	3210h
	IPR1	0x3002A	Interrupt priority register 1	R/W	7654h
	IPR2	0x3002C	Interrupt priority register 2	R/W	BA98h
	IPR3	0x3002E	Interrupt priority register 3	R/W	FEDCH
I/O ports	P0	0x30030	Port0 data register	R/W	xx00h
	P1	0x30031	Port1 data register	R/W	00xxh
	P2	0x30032	Port2 data register	R/W	xx00h
	P3	0x30033	Port3 data register	R/W	00xxh
	P0CON	0x30034	Port0 control register	R/W	0000h
	P1CON	0x30036	Port1 control register	R/W	0000h
	P2CON	0x30038	Port2 control register	R/W	0000h
	P3CON	0x3003A	Port3 control register	R/W	xxx0h

Table 5-1. Special Registers (Continued)

Group	Registers	Address	Description	R/W	Reset Value
ADC	ADCON	0x3003C	A/D conversion mode register	R/W	x000h
–	TSTC (note)	0x3003E	Test control register	–	–
On screen display	OSGM3	0x30040	OSD graphic color mode 3	R/W	0000h
	OSGM2	0x30042	OSD graphic color mode 2	R/W	0000h
	OSGM1	0x30044	OSD graphic color mode 1	R/W	0000h
	OSGDATA	0x30046	OSD graphic color data	R/W	0000h
	OSDPLTR	0x30048	OSD palette color mode R	R/W	FF00h
	OSDPLTG	0x3004A	OSD palette color mode G	R/W	F0F0h
	OSDPLTB	0x3004C	OSD palette color mode B	R/W	CCCCh
	OSDFLD	0x3004E	OSD field control register	R/W	0006h
	OSDCON	0x30050	OSD control register	R/W	0000h
	FADECON	0x30052	OSD FADE control register	R/W	0000h
	OSDBGD	0x30054	OSD background color register	R/W	0000h
	OSDVMGN	0x30056	OSD margin control register	R/W	2400h
	OSDCNT	0x30058	OSD counter register	R/W	xxxxh
Location 0x3005A - 0x3005F are not mapped					
Interrupt	IRQV0	0x30060	IRQ vector register0	R/W	0000h
	IRQV1	0x30062	IRQ vector register1	R/W	0000h
	IRQV2	0x30064	IRQ vector register2	R/W	0000h
	IRQV3	0x30066	IRQ vector register3	R/W	0000h
	IRQV4	0x30068	IRQ vector register4	R/W	0000h
	IRQV5	0x3006A	IRQ vector register5	R/W	0000h
	IRQV6	0x3006C	IRQ vector register6	R/W	0000h
	IRQV7	0x3006E	IRQ vector register7	R/W	0000h
	IRQV8	0x30070	IRQ vector register8	R/W	0000h
	IRQV9	0x30072	IRQ vector register9	R/W	0000h
	IRQV10	0x30074	IRQ vector register10	R/W	0000h
	IRQV11	0x30076	IRQ vector register11	R/W	0000h
	IRQV12	0x30078	IRQ vector register12	R/W	0000h
	IRQV13	0x3007A	IRQ vector register13	R/W	0000h
	IRQV14	0x3007C	IRQ vector register14	R/W	0000h
	IRQV15	0x3007E	IRQ vector register15	R/W	0000h

NOTE: TSTC is used for factory test only. TSTC must not be accessed by program.

Table 5-1. Special Registers (Continued)

Group	Registers	Offset	Description	R/W	Reset Value
Interrupt	FIQV0	0x30080	FIQ vector register0	R/W	0000h
	FIQV1	0x30082	FIQ vector register1	R/W	0000h
	FIQV2	0x30084	FIQ vector register2	R/W	0000h
	FIQV3	0x30086	FIQ vector register3	R/W	0000h
	FIQV4	0x30088	FIQ vector register4	R/W	0000h
	FIQV5	0x3008A	FIQ vector register5	R/W	0000h
	FIQV6	0x3008C	FIQ vector register6	R/W	0000h
	FIQV7	0x3008E	FIQ vector register7	R/W	0000h
	FIQV8	0x30090	FIQ vector register8	R/W	0000h
	FIQV9	0x30092	FIQ vector register9	R/W	0000h
	FIQV10	0x30094	FIQ vector register10	R/W	0000h
	FIQV11	0x30096	FIQ vector register11	R/W	0000h
	FIQV12	0x30098	FIQ vector register12	R/W	0000h
	FIQV13	0x3009A	FIQ vector register13	R/W	0000h
	FIQV14	0x3009C	FIQ vector register14	R/W	0000h
	FIQV15	0x3009E	FIQ vector register15	R/W	0000h
Location 0x300A0 - 0x300A7 are not mapped					
Interrupt	IRQSR	0x300A8	IRQ shadow register	R	xxxxxxxxh
	FIQSR	0x300AC	FIQ shadow register	R	xxxxxxxxh

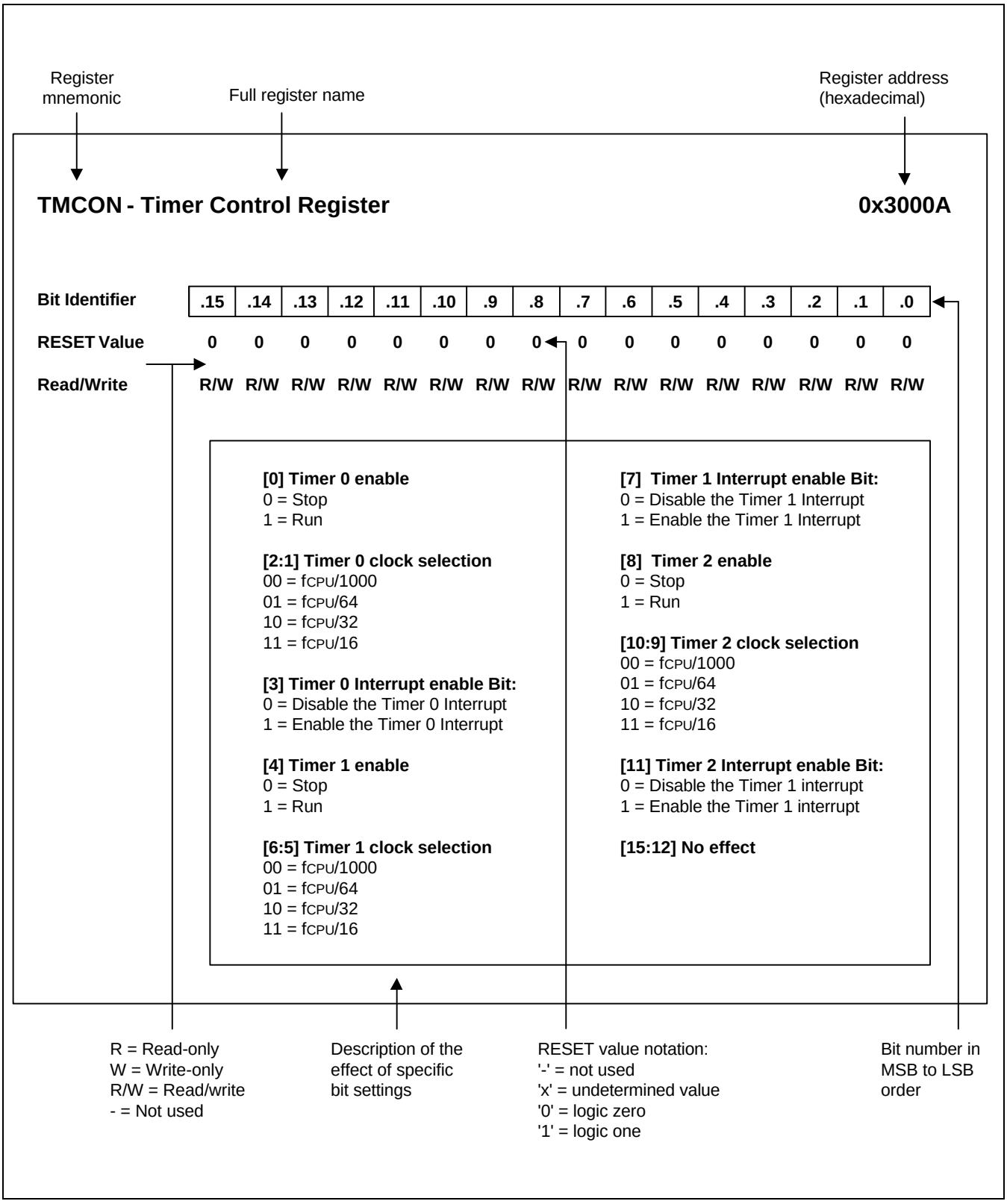


Figure 5-1. Register Description Format

ADCON - A/D CONVERSION MODE REGISTER 0x3003C

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	–	0	0	0	0	0	0	0	0	0	0
Read/Write	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

[3:0] A/D conversion Data (Read-only)

[4] Flash ADC Mode Selection
0 = Flash ADC0 mode
1 = Flash ADC1-4 mode

[6:5] A/D Conversion Selection
00 = Select ADC1
01 = Select ADC2
10 = Select ADC3
11 = Select ADC4

[7] A/D Conversion Control Bit
0 = Stop A/D conversion
1 = Start A/D conversion

[9:8] A/D Conversion Time Selection
00 = $f_{CPU}/4$
01 = $f_{CPU}/8$
10 = $f_{CPU}/16$
11 = $f_{CPU}/32$

[15:10] Not Used

BTCON - BASIC TIMER CONTROL REGISTER 0x30000

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0	–	–	–	–	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	–	–	–	R/W	R/W	R/W	R/W

[0] Watchdog Counter (WTCNT) Clear Bit
0 = No effect
1 = Watchdog counter clear

[1] Basic Counter (BTCNT) Clear Bit
0 = No effect
1 = Basic counter clear

[3:2] Clock Source Select
00 = $X_{IN}/2^4$
01 = $X_{IN}/2^3$
10 = $X_{IN}/2^2$
11 = X_{IN}

[7:4] No Effect

[15:8] Watchdog Timer Enable
10100101 = Disable
Others = Enable

BTCNT - BASIC TIMER COUNTER REGISTER 0x30002

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	–	–	–	0	0	0	0	0	0	0	0
Read/Write	–	–	–	–	–	–	–	–	R	R	R	R	R	R	R	R

[7:0] Basic Timer Counter Value (8 bit)
[15:8] No Effect

FADECON - OSD FADE CONTROL REGISTER 0x30052

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[5:0] Fade Line Address (0-17h) + Inter ROW Space (15h)
000000 = Line 0
000001 = Line 1
000010 = Line 2
000011 = Line 3
000100 = Line 4
000101 = Line 5
000110 = Line 6
000111 = Line 7
001000 = Line 8
001001 = Line 9
001010 = Line 10
001011 = Line 11
001100 = Line 12
001101 = Line 13
001110 = Line 14
001111 = Line 15
010000 = Line 16
010001 = Line 17
010010 = Inter row space line 1
010011 = Inter row space line 2
010100 = Inter row space line 3
010101 = Inter row space line 4
010110 = Inter row space line 5
010111 = Inter row space line 6
011000 = Inter row space line 7
011001 = Inter row space line 8
011010 = Inter row space line 9
011011 = Inter row space line 10
011100 = Inter row space line 11
011101 = Inter row space line 12
011110 = Inter row space line 13
011111 = Inter row space line 14
100000 = Inter row space line 15
100001 = Not used
111111 = Not used

[6] Fade Direction Selection Bit
0 = Fade before matrix
1 = Fade after matrix

[7] Fade Function Enable Bit
0 = Fade disable
1 = Fade enable

[11:8] Fade Row Address Selection
0000 = Row 0
0001 = Row 1
0010 = Row 2
0011 = Row 3
0100 = Row 4
0101 = Row 5
0110 = Row 6
0111 = Row 7
1000 = Row 8
1001 = Row 9
1010 = Row 10
1011 = Row 11
1100 = Row 12
1101 = Row 13
1110 = Row 14
1111 = Row 15

[15:12] No Effect

FIFOD - FIFO DATA REGISTER 0x3001A

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	–	–	–	x	x	x	x	x	x	x	x
Read/Write	–	–	–	–	–	–	–	–	R	R	R	R	R	R	R	R

[7:0] FIFO Data (8-Bit Count Value)
[15:8] No Effect

INTPR - INTERRUPT PENDING REGISTER 0x30022

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[0] 2nd edge H-sync interrupt
[1] 1st edge H-sync interrupt
[2] OSD row interrupt
[3] Timer 0 interrupt
[4] Timer 1 interrupt
[5] Timer 2 interrupt
[6] Remote control signal input interrupt (Capture)
[7] FIFO full interrupt
[8] Remocon counter overflow interrupt
[9] V-sync interrupt
[10] Basic timer interrupt
[11] External interrupt 0
[12] External interrupt 1
[13] External interrupt 2
[14] External interrupt 3
[15] Real timer interrupt

INTMD - INTERRUPT MODE REGISTER 0x30024

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[0] 2nd edge H-sync interrupt
[1] 1st edge H-sync interrupt
[2] OSD row interrupt
[3] Timer 0 interrupt
[4] Timer 1 interrupt
[5] Timer 2 interrupt
[6] Remote control signal input interrupt (Capture)
[7] FIFO full interrupt
[8] Remocon counter overflow interrupt
[9] V-sync interrupt
[10] Basic timer interrupt
[11] External interrupt 0
[12] External interrupt 1
[13] External interrupt 2
[14] External interrupt 3
[15] Real timer interrupt

INTMK - INTERRUPT MASK REGISTER 0x30026

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[0] 2nd edge H-sync interrupt
[1] 1st edge H-sync interrupt
[2] OSD row interrupt
[3] Timer 0 interrupt
[4] Timer 1 V-sync interrupt
[5] Timer 2 interrupt
[6] Remote control signal input interrupt (Capture)
[7] FIFO full interrupt
[8] Remocon counter overflow interrupt
[9] V-sync interrupt
[10] Basic timer interrupt
[11] External interrupt 0
[12] External interrupt 1
[13] External interrupt 2
[14] External interrupt 3
[15] Real timer interrupt

IPR0 - INTERRUPT PRORITY REGISTER 0 0x30028

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[3:0] Priority 0 Setting
[7:4] Priority 1 Setting
[11:8] Priority 2 Setting
[15:12] Priority 3 Setting

IPR1 - INTERRUPT PRORITY REGISTER 1 0x3002A

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[3:0] Priority 4 Setting
[7:4] Priority 5 Setting
[11:8] Priority 6 Setting
[15:12] Priority 7 Setting

IPR2 - INTERRUPT PRORITY REGISTER 2 0x3002C

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	1	0	1	1	1	0	1	0	1	0	0	1	1	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[3:0] Priority 8 Setting
[7:4] Priority 9 Setting
[11:8] Priority 10 Setting
[15:12] Priority 11 Setting

IPR3 - INTERRUPT PRORITY REGISTER 3 0x3002E

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	1	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[3:0] Priority 12 Setting
[7:4] Priority 13 Setting
[11:8] Priority 14 Setting
[15:12] Priority 15 Setting

OSDBGD - OSD BACKGROUND COLOR REGISTER 0x30054

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0	–	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[3:0] Frame Background
0xxx = No frame background
1000 = Color mode 0
1001 = Color mode 1
1010 = Color mode 2
1011 = Color mode 3
1100 = Color mode 4
1101 = Color mode 5
1110 = Color mode 6
1111 = Color mode 7

[4] All Frame H/T Function
0 = Disable
1 = Enable

[5] Inter-row Space Color
0 = Same as character background color
1 = Same as frame background color

[6] Inter-tow Space Half Tone
0 = Disable
1 = Enable

[7] No Effect

[10:8] Inter-column Spacing Control (0-7 OSD Clock)
000 = No inter-column spacing
001 = Inter-column spacing = 1 OSD clock
010 = Inter-column spacing = 2 OSD clock
011 = Inter-column spacing = 3 OSD clock
100 = Inter-column spacing = 4 OSD clock
101 = Inter-column spacing = 5 OSD clock
110 = Inter-column spacing = 6 OSD clock
111 = Inter-column spacing = 7 OSD clock

[15:11] Left Margin Display Position Control (16 + (0-32 Dot)
00000 = Left margin (16 OSD clocks)
00001 = Left margin (16 + 1 OSD clocks)
00010 = Left margin (16 + 2 OSD clocks)
. .
. .
. .
11110 = Left margin (16 + 30 OSD clocks)
11111 = Left margin (16 + 31 OSD clocks)

OSDCNT - OSD COUNTER REGISTER 0x30058

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	x	x	x	x	–	–	x	x	x	x	x	x
Read/Write	–	–	–	–	R/W	R/W	R/W	R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W

[5:0] Line Count Value
000000 = 1st line
000001 = 2nd line
000010 = 3rd line
000011 = 4th line
.
.
.
111110 = 63rd line
111111 = 64th line

[11:8] Row Count Value
0000 = Row 0
0001 = Row 1
0010 = Row 2
0011 = Row 3
0100 = Row 4
0101 = Row 5
0110 = Row 6
0111 = Row 7
1000 = Row 8
1001 = Row 9
1010 = Row 10
1011 = Row 11
1100 = Row 12
1101 = Row 13
1110 = Row 14
1111 = Row 15

[15:12] Not Used

NOTE: H-sync count value is cleared by v-sync input and line 0 of every row (same as OSD row interrupt).

OSDCON - OSD CCD DISPLAY CONTROL REGISTER 0x30050

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	–	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[0] OSD/CCD Mode
0 = OSD mode
1 = CCD mode

[2:1] Horizontal Character Size
00 = 1 size
01 = 2 size
10 = 3 size
11 = 4 size

[4:3] Vertical Character Size
00 = 1 size
01 = 2 size
10 = 3 size
11 = 4 size

[5] OSD RGB Polarity
0 = Positive signal output
1 = Negative signal output

[6] Halftone Polarity
0 = Active high level
1 = Active low level

[7] H/V-sync polarity
0 = External mode (rising edges)
1 = External mode (falling edges)

[8] Italic Character
0 = Italic character disable
1 = Italic character enable

[9] V-sync Interrupt Enable Bit
0 = Disable v-sync interrupt
1 = Enable v-sync interrupt

[10] H-sync Interrupt Enable Bit
0 = Disable h-sync interrupt
1 = Enable h-sync interrupt

[11] OSD Row Interrupt Enable Bit
0 = Disable OSD row interrupt
1 = Enable OSD row interrupt

[12] No Effect

[13] Line Graphic Enable Bit
0 = Disable line graphic
1 = Enable line graphic

[14] Graphic OSD Enable Bit
0 = Disable graphic OSD (video RAM (01H): OSD code)
1 = Enable graphic OSD (video RAM (01H): graphic code)

[15] OSD/CCD Display Enable Bit
0 = Disable OSD/CCD
1 = Enable OSD/CCD

OSDFLD - OSD FIELD CONTROL REGISTER 0x3004E

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	–	–	–	–	–	x	0	0	0	0	0
Read/Write	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W

[3:0] Even Field Range
0000 = Not used
0001 = $f_{VCO}/128 \times 1$
0010 = $f_{VCO}/128 \times 2$
0011 = $f_{VCO}/128 \times 3$
0100 = $f_{VCO}/128 \times 4$
0101 = $f_{VCO}/128 \times 5$
0110 = $f_{VCO}/128 \times 6$ (reset value)
0111 = $f_{VCO}/128 \times 7$
1000 = $f_{VCO}/128 \times 8$
1001 = $f_{VCO}/128 \times 9$
1010 = $f_{VCO}/128 \times 10$
1011 = $f_{VCO}/128 \times 11$
1100 = $f_{VCO}/128 \times 12$
1101 = $f_{VCO}/128 \times 13$
1110 = $f_{VCO}/128 \times 14$
1111 = $f_{VCO}/128 \times 15$

[4] H-sync Detect Position Select
0 = Detect H-sync before V-sync
1 = Detect H-sync after V-sync

[5] Field Data (Read Only)
0 = Even field
1 = Odd field

[15:6] No Effect

OSDPLTB - OSD PALETTE COLOR MODE B 0x3004C

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1:0] OSD Mode 0 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[3:2] OSD Mode 1 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[5:4] OSD Mode 2 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[7:6] OSD Mode 3 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[9:8] OSD Mode 4 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[11:10] OSD Mode 5 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[13:12] OSD Mode 6 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[15:14] OSD Mode 7 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

OSDPLTG - OSD PALETTE COLOR MODE G 0x3004A

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1:0] OSD Mode 0 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[3:2] OSD Mode 1 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[5:4] OSD Mode 2 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[7:6] OSD Mode 3 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[9:8] OSD Mode 4 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[11:10] OSD Mode 5 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[13:12] OSD Mode 6 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[15:14] OSD Mode 7 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

OSDPLTR - OSD PALETTE COLOR MODE R 0x30048

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1:0] OSD Mode 0 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[3:2] OSD Mode 1 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[5:4] OSD Mode 2 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[7:6] OSD Mode 3 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[9:8] OSD Mode 4 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[11:10] OSD Mode 5 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[13:12] OSD Mode 6 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[15:14] OSD Mode 7 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

OSDVMGN - OSD MARGIN CONTROL REGISTER 0x30056

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Read/Write	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[3:0] Inter-row Space Control (H-sync Pulses)
0000 = No inter-row space (0H)
0001 = 1H inter-row space
0010 = 2H inter-row space
0011 = 3H inter-row space
0100 = 4H inter-row space
. .
. .
. .
1101 = 13H inter-row space
1110 = 14H inter-row space
1111 = 15H inter-row space

[9:4] Top Margin Display Position Control (1H-64H)
000000 = Top margin (1H)
000001 = Top margin (2H)
000010 = Top margin (3H)
. .
. .
. .
111110 = Top margin (63H)
111111 = Top margin (64H)

[14:10] V-sync Blank Control Register (Reset: 9 Horizontal Sync)
00000–01001 = 9 H-sync
01010 = 10 H-sync
01011 = 11 H-sync
01100 = 12 H-sync
. .
. .
. .
11110 = 30 h-sync
11111 = 31 h-sync

[15] No Effect

P0CON - PORT0 CONTROL REGISTER 0x30034

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1:0] P0.0/PWM 0 Setting
0x = Input mode
10 = Output mode (push-pull type)
11 = Alternative function mode (PWM0 function output; push-pull type)

[3:2] P0.1/PWM 1 Setting
0x = Input mode
10 = Output mode (open-drain type)
11 = Alternative function mode (PWM1 function output; open-drain type)

[5:4] P0.2 Setting
0x = Input mode
10 = Not used
11 = Output mode (open-drain type)

[7:6] P0.3 Setting
0x = Input mode
10 = Not used
11 = Output mode (open-drain type)

[9:8] P0.4 Setting
00 = Input mode
01 = Input mode
10 = Output mode (open-drain type)
11 = Output mode (push-pull type)

[11:10] P0.5 Setting
00 = Input mode
01 = Input mode
10 = Output mode (open-drain type)
11 = Output mode (push-pull type)

[13:12] P0.6 Setting
00 = Input mode
01 = Input mode
10 = Output mode (open-drain type)
11 = Output mode (push-pull type)

[15:14] P0.7 Setting
00 = Input mode
01 = Input mode
10 = Output mode (open-drain type)
11 = Output mode (push-pull type)

P1CON - PORT1 CONTROL REGISTER 0x30036

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[0:1] P1.0 Setting
00 = Input mode
01–10 = Not used
11 = Output mode (push-pull type)

[3:2] P1.1 Setting
00 = Input mode
01–10 = Not used
11 = Output mode (push-pull type)

[5:4] P1.2 Setting
00 = Input mode
01–10 = Not used
11 = Output mode (push-pull type)

[7:6] P1.3 Setting
00 = Input mode
01–10 = Not used
11 = Output mode (push-pull type)

[9:8] P1.4/ADC1 Setting
00 = Input mode
01 = ADC input mode (digital input disable)
10 = Not used
11 = N-channel open-drain output mode (5 V load capacity)

[11:10] P1.5/ADC2 Setting
00 = Input mode
01 = ADC input mode (digital input disable)
10 = Not used
11 = N-channel open-drain output mode (5 V load capacity)

[13:12] P1.6/ADC3 Setting
00 = Input mode
01 = ADC input mode (digital input disable)
10 = Not used
11 = N-channel open-drain output mode (5 V load capacity)

[15:14] P1.7/ADC4 Setting
00 = Input mode
01 = ADC input mode (digital input disable)
10 = Not used
11 = N-channel open-drain output mode (5 V load capacity)

P2CON - PORT2 CONTROL REGISTER 0x30038

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1:0] P2.0/INT0 Setting
00 = Input mode, interrupt disabled
01 = Input mode, interrupt on rising edge
10 = Input mode, interrupt on falling edge
11 = Output mode (push-pull type)

[3:2] P2.1/INT1 Setting
00 = Input mode, interrupt disabled
01 = Input mode, interrupt on rising edge
10 = Input mode, interrupt on falling edge
11 = Output mode (push-pull type)

[5:4] P2.2/INT2 Setting
00 = Input mode, interrupt disabled
01 = Input mode, interrupt on rising edge
10 = Input mode, interrupt on falling edge
11 = Output mode (push-pull type)

[7:6] P2.3/INT3 Setting
00 = Input mode, interrupt disabled
01 = Input mode, interrupt on rising edge
10 = Input mode, interrupt on falling edge
11 = Output mode (push-pull type)

[9:8] P2.4 Setting
00 = Input mode
01–10 = Not used
11 = Output mode (push-pull type)

[11:10] P2.5 Setting
00 = Input mode
01–10 = Not used
11 = Output mode (push-pull type)

[13:12] P2.6 Setting
00 = Input mode
01–10 = Not used
11 = Output mode (push-pull type)

[15:14] P2.7/OSDHT Setting
00 = Input mode
01 = Not used
10 = OSDHT output mode
11 = Output mode (push-pull type)

P3CON - PORT3 CONTROL REGISTER 0x3003A

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1:0] P3.0 Setting
00 = Input mode
01–10 = Not used
11 = Output mode (push-pull type)

[15:2] No Effect

PLLCON - PLL CLOCK CONTROL REGISTER 0x3001E

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	1	0	1	1	0	1	1	1	–	1	1	0	0	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[2:0] CPU Clock Selection
000 = No effect
001 = $f_{VCO}/2$
010 = $f_{VCO}/3$
011 = $f_{VCO}/4$
100 = $f_{VCO}/5$
101 = $f_{VCO}/6$
110 = $f_{VCO}/7$
111 = $f_{VCO}/8$

[3] Operating Mode Selection
0 = Normal operation mode
1 = SLEEP mode

[6:4] OSD Clock Selection
000 = No effect
001 = $f_{VCO}/2$
010 = $f_{VCO}/3$
011 = $f_{VCO}/4$
100 = $f_{VCO}/5$
101 = $f_{VCO}/6$
110 = $f_{VCO}/7$
111 = $f_{VCO}/8$

[7] No Effect

[15:8] f_{VCO} Clock Selection
10010110 = 96H, 40.00 MHz
10010111 = 97H, 40.25 MHz
10011000 = 98H, 40.50 MHz
10011001 = 99H, 40.75 MHz
10011010 = 9AH, 41.00 MHz
10011011 = 9BH, 41.25 MHz
. .
. .
. .
11001001 = C9H, 52.75 MHz
11001010 = CAH, 53.00 MHz
11001011 = CBH, 53.25 MHz
11001100 = CCH, 53.50 MHz
11001101 = CDH, 53.75 MHz
11001110 = CEH, 54.00 MHz

NOTES:
1. f_{VCO} default value = B7H (48.25 MHz)
2. f_{CPU} default value = 3H (12.06 MHz)
3. f_{OSD} default value = 6H (6.89 MHz)

PWM0 - PWM0 DATA REGISTER (14-BIT) 0x30014

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	1	1	1	1	1	1	1	1	0	0	0	0	0	0
Read/Write	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[5:0] PWM0 Extension (Stretch) Value Set
000000 = No effect
000001 = 32
000010 = 16, 48
000100 = 8, 24, 40, 56
001000 = 4, 12, 20, …, 44, 52, 60
010000 = 2, 6, 10, 14, …, 50, 54, 58, 62
100000 = 1, 3, 5, 7, 9, 11, …, 55, 57, 59, 61, 63

[13:6] PWM0 Basic Cycle Frame Value Set

[15:14] No Effect

PWM1 - PWM1 DATA REGISTER (14-BIT) 0x30016

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	1	1	1	1	1	1	1	1	0	0	0	0	0	0
Read/Write	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[5:0] PWM1 Extension (Stretch) Value Set
000000 = No effect
000001 = 32
000010 = 16, 48
000100 = 8, 24, 40, 56
001000 = 4, 12, 20, …, 44, 52, 60
010000 = 2, 6, 10, 14, …, 50, 54, 58, 62
100000 = 1, 3, 5, 7, 9, 11, …, 55, 57, 59, 61, 63

[13:6] PWM1 Basic Cycle Frame Value Set

[15:14] No Effect

PWMCON - PWM CONTROL REGISTER 0x30012

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	–	–	–	–	–	–	0	0	0	0	0
Read/Write	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W

[3:0] PWM0–PWM1 Prescaler
0000 = Non divider
0001 = Divider by 2
0010 = Divider by 3
0011 = Divider by 4
0100 = Divider by 5
0101 = Divider by 6
0110 = Divider by 7
0111 = Divider by 8
1000 = Divider by 9
1001 = Divider by 10
1010 = Divider by 11
1011 = Divider by 12
1100 = Divider by 13
1101 = Divider by 14
1110 = Divider by 15
1111 = Divider by 16

[4] PWM0–PWM1 Counter Enable Bit
0 = Disable
1 = Enable

[15:5] No Effect

RRCR - REMOCON RECEIVE CONTROL REGISTER 0x30018

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1:0] Polarity Control Flag
00 = Not used
01 = Rising edge mode
10 = Falling edge mode
11 = Rising/falling edge mode

[3:2] Minimum pulse width recognized as remocon signal
00 = f_{VCO}/128
01 = f_{VCO}/256
10 = f_{VCO}/512
11 = 32,768 Hz

[5:4] Counter Clock Selection
00 = 32,768 Hz/1 (30.52 μsec)
01 = 32,768 Hz/2 (61.03 μsec)
10 = 32,768 Hz/4 (122.07 μsec)
11 = 32,768 Hz/8 (244.14 μsec)

[7:6] No Effect

[8] FIFO Empty Status Flag
0 = FIFO not empty
1 = FIFO empty

[9] FIFO Full Status Flag
0 = FIFO not full
1 = FIFO full

[10] Remocon Input Interrupt
0 = Disable the remocon input interrupt
1 = Enable the remocon input interrupt

[11] FIFO Full(8)-step Detect Interrupt
0 = Disable the FIFO full(8)-step detect interrupt
0 = Enable the FIFO full(8)-step detect interrupt

[12] Counter Overflow Interrupt
0 = Disable the counter overflow interrupt
1 = Enable the counter overflow interrupt

[15:13] No Effect

NOTE: If FIFO full, next input data is not come into FIFO.

RTCON - REAL TIME CLOCK CONTROL REGISTER 0x3001C

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	–	–	–	–	–	–	–	–	0	0	0
Read/Write	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W

[0] Real Time Clock Counter Clear Bit:
0 = No effect
1 = Real time clock counter clear

[1] Real Time Clock Enable Bit:
0 = Disable
1 = Enable

[2] Interrupt (1sec) Enable Bit:
0 = Disable
1 = Enable

[15:3] No Effect

SYSCON - SYSTEM CONTROL REGISTER 0x30020

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	–	–	–	–	–	–	–	–	0	0	0
Read/Write	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W

[0] Clock Wait for Fast ROM
0 = Wait enable (2 clock)
1 = Wait disable (1 clock)

[1] Shadow Register Enable
0 = Disable
1 = Enable

[2] Instruction Prefetch
0 = Disable
1 = Enable

[15:3] No Effect

TMCON - TIMER CONTROL REGISTER 0x3000A

Bit Identifier	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[0] Timer 0 Enable
0 = Stop
1 = Run

[2:1] Timer 0 Clock Selection
00 = f_{CLK}/1000
01 = f_{CLK}/64
10 = f_{CLK}/32
11 = f_{CLK}/16

[3] Timer 0 Interrupt Enable Bit:
0 = Disable the timer 0 interrupt
1 = Enable the timer 0 interrupt

[4] Timer 1 Enable
0 = Stop
1 = Run

[6:5] Timer 1 Clock Selection
00 = f_{CLK}/1000
01 = f_{CLK}/64
10 = f_{CLK}/32
11 = f_{CLK}/16

[7] Timer 1 Interrupt Enable Bit:
0 = Disable the timer 1 interrupt
1 = Enable the timer 1 interrupt

[8] Timer 2 Enable
0 = Stop
1 = Run

[10:9] Timer 2 Clock Selection
00 = f_{CLK}/1000
01 = f_{CLK}/64
10 = f_{CLK}/32
11 = f_{CLK}/16

[11] Timer 2 Interrupt Enable Bit:
0 = Disable the timer 2 interrupt
1 = Enable the timer 2 interrupt

[15:12] No Effect

6 SYSTEM RESET and POWER-DOWN MODE

SYSTEM RESET

OVERVIEW

During a power-on reset, the voltage at V_{DD} goes to High level and the RESET pin is forced to Low level. The RESET signal is input through a Schmitt trigger circuit where it is then synchronized with the CPU clock.

To allow time for internal CPU clock oscillation to stabilize, the RESET pin must be held to Low level for a minimum time interval after the power supply comes within tolerance. The minimum required oscillation stabilization time for a reset operation is 500 millisecond.

Whenever a reset occurs during normal operation (that is, when both V_{DD} and RESET are in High level), the RESET pin is forced into Low and the reset operation starts. All system and peripheral control registers are then reset to their default hardware values (see Tables 6-1, 6-2, 6-3, and 6-4).

In summary, the following sequence of events occurs during a reset operation:

— All interrupts are disabled.

— The watchdog function (basic timer) is enabled.

— Ports 0-3 are set to input mode.

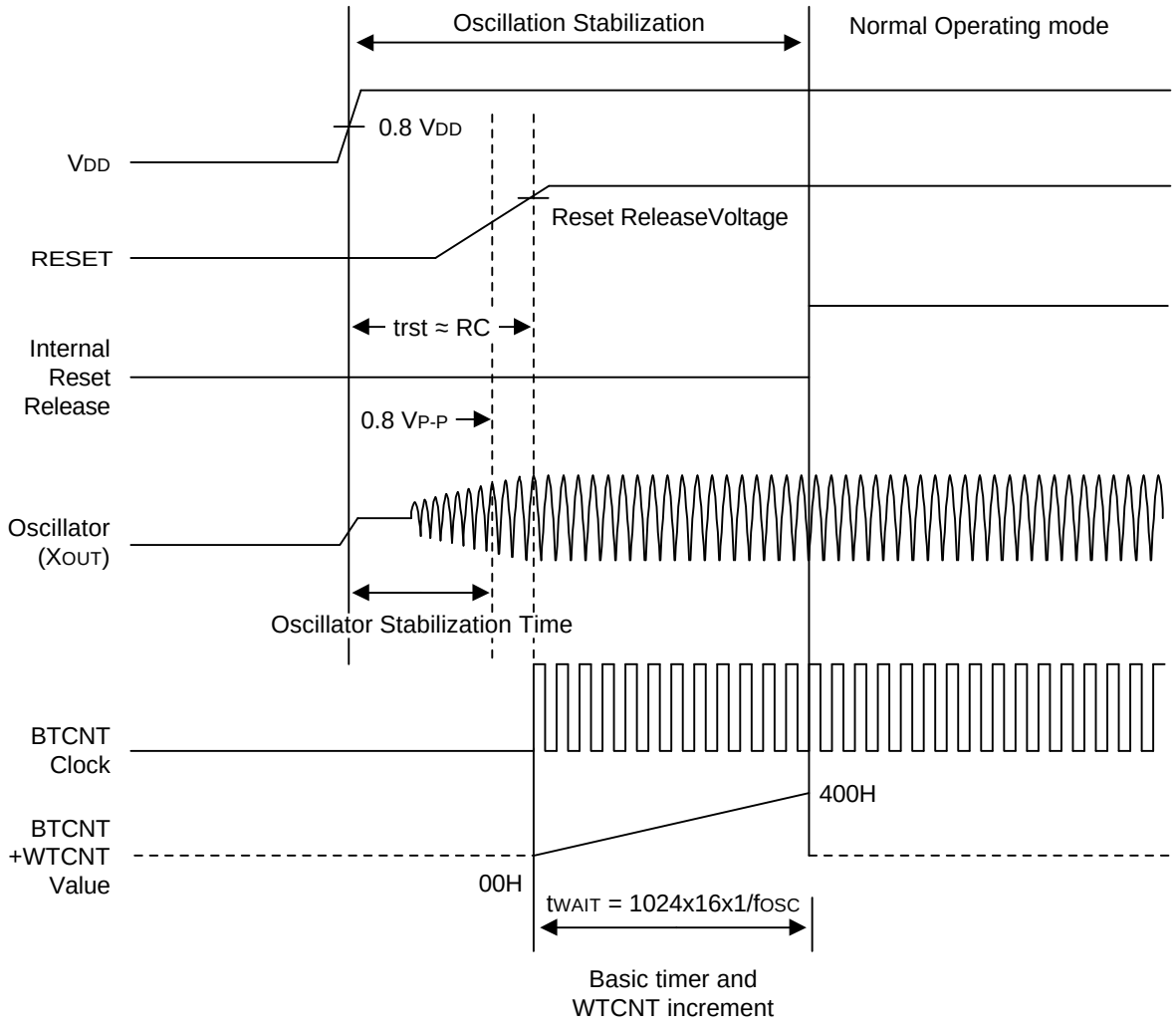
— Peripheral control and data registers are disabled and reset to their default hardware values.

— The program counter (PC) is loaded with the program reset address in the ROM, 0x0000.

— When the programmed oscillation stabilization time interval has elapsed, the instruction stored in ROM location 0x0000 (and 0x0001, 0x0002, 0x0003) is fetched and executed.

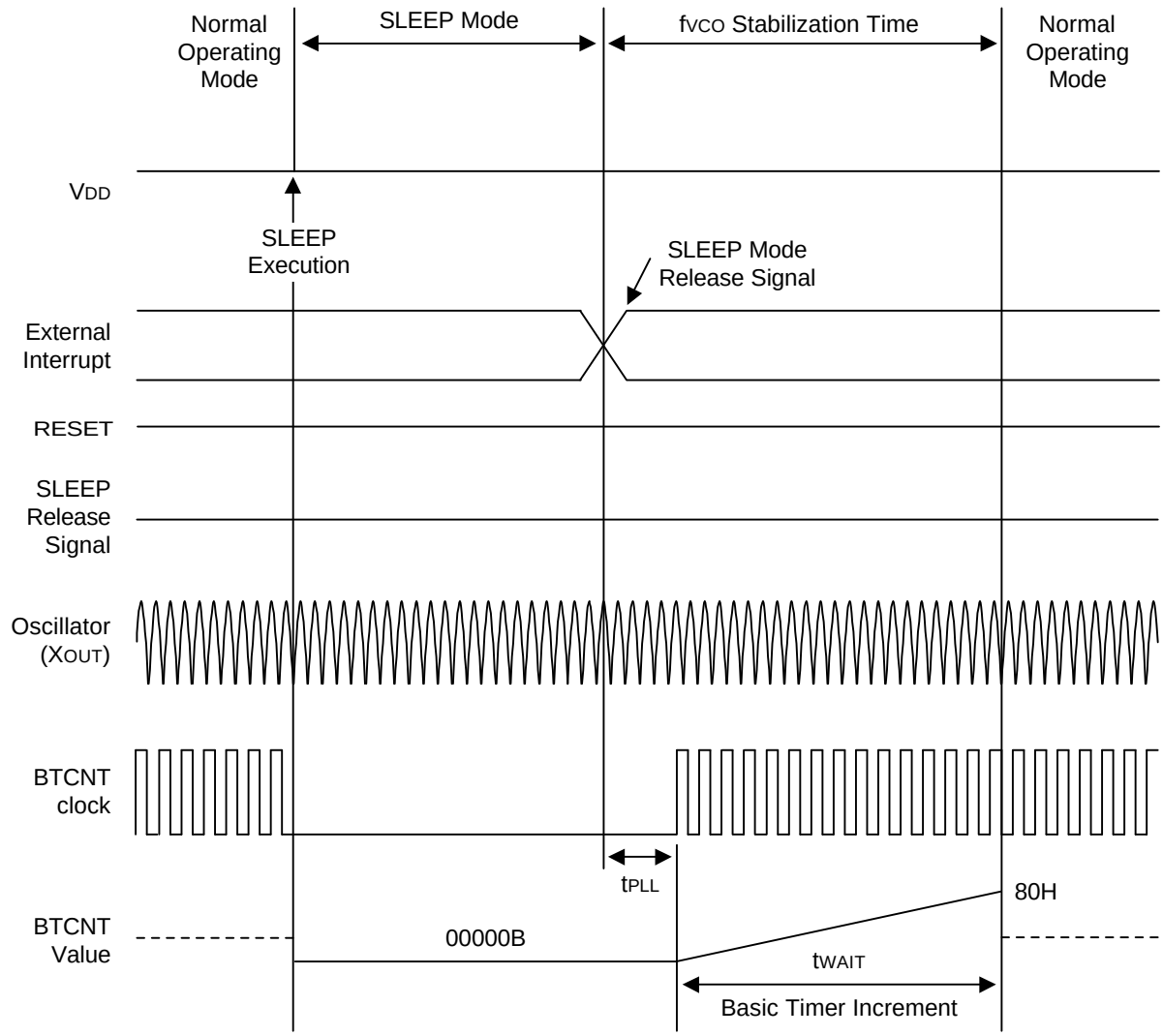
NOTE

If you do not want to use the basic timer watchdog function (which causes a system reset if a basic timer counter overflow occurs), you can disable it by writing '10100101B' to the bit 15-8 of BTCON before entering sleep mode.



NOTE: After a power-on reset, the oscillator stabilization wait time, tWAIT, takes 1024x16/fosc. (500 ms at fOSC = 32,768 Hz)
trst ≈ RC (R is an on-chip resistor and C is an external capacitor.

Figure 6-1. Oscillation Stabilization Time on RESET



NOTE: After a release by an interrupt, the oscillator stabilization wait time, t_{WAIT} takes 128x1/f_{OSC}. (4ms at f_{OSC} = 32,768 Hz)
t_{PLL}: PLL lock time

Figure 6-2. Oscillation Stabilization Time on Sleep Mode Release

HARDWARE RESET VALUES

Tables 6-1 through 6-4 list the reset values for CPU and system registers, peripheral control registers, and peripheral data registers after a reset operation. The following notation is used to represent reset values:

— A "1" or "0" shows the reset bit value as logic one or logic zero, respectively.

— An 'x' means that the bit value is undefined after a reset.

— A dash ('–') means that the bit is either not used or not mapped.

Table 6-1. Set 1 Register Values after a Reset

Register Name	Mnemonic	Address	Bit Values after a Reset															
		Hex	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Watchdog timer control register	BTCON	0x30000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Basic timer counter register	BTCNT	0x30002	–	–	–	–	–	–	–	–	0	0	0	0	0	0	0	0
Timer 0 data register	T0DATA	0x30004	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Timer 1 data register	T1DATA	0x30006	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Timer 2 data register	T2DATA	0x30008	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Timer control register	TMCON	0x3000A	–	–	–	–	0	0	0	0	0	0	0	0	0	0	0	0
Timer 0 counter register	T0CNT	0x3000C	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Timer 1 counter register	T1CNT	0x3000E	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Timer 2 counter register	T2CNT	0x30010	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
PWM control register	PWMCON	0x30012	–	–	–	–	–	–	–	–	–	–	–	0	0	0	0	0
PWM 0 data register (14Bit)	PWM0	0x30014	–	–	1	1	1	1	1	1	1	1	0	0	0	0	0	0
PWM 1 data register (14Bit)	PWM1	0x30016	–	–	1	1	1	1	1	1	1	1	0	0	0	0	0	0
Remocon receive control register	RRCR	0x30018	–	–	–	0	0	0	0	1	0	0	0	0	0	0	0	0
FIFO data register	FIFOD	0x3001A	–	–	–	–	–	–	–	–	x	x	x	x	x	x	x	x
Real time clock control register	RTCON	0x3001C	–	–	–	–	–	–	–	–	–	–	–	–	–	0	0	0
PLL clock control register	PLLCON	0x3001E	1	0	1	1	0	1	1	1	–	1	1	0	0	0	1	1

Table 6-1. Set 1 Register Values after a Reset (Continued)

Register Name	Mnemonic	Address	Bit Values after a Reset															
		Hex	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
System control register	SYSCON	0x30020	–	–	–	–	–	–	–	–	–	–	–	–	–	0	0	0
Interrupt pending register	INTPR	0x30022	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Interrupt mode register	INTMD	0x30024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Interrupt mask register	INTMK	0x30026	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Interrupt priority register 0	IPR0	0x30028	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0
Interrupt priority register 1	IPR1	0x3002A	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0
Interrupt priority register 2	IPR2	0x3002C	1	0	1	1	1	0	1	0	1	0	0	1	1	0	0	0
Interrupt priority register 3	IPR3	0x3002E	1	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0
Port0 data register	P0	0x30030	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0
Port1 data register	P1	0x30031	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x
Port2 data register	P2	0x30032	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0
Port3 data register	P3	0x30033	–	–	–	–	–	–	–	0	x	x	x	x	x	x	x	x
Port0 control register	P0CON	0x30034	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Port1 control register	P1CON	0x30036	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Port2 control register	P2CON	0x30038	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Port3 control register	P3CON	0x3003A	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0	0
A/D conversion mode register	ADCON	0x3003C	–	–	–	–	–	–	0	0	0	0	0	0	0	0	0	0
OSD Graphic Color Mode 3	OSGM3	0x30040	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OSD Graphic Color Mode 2	OSGM2	0x30042	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6-1. Set 1 Register Values after a Reset (Continued)

Register Name	Mnemonic	Address	Bit Values after a Reset															
		Hex	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSD Graphic Color Mode 1	OSGM1	0x30044	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OSD Graphic Color Data	OSGDATA	0x30046	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OSD Palette Color Mode R	OSDPLTR	0x30048	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
OSD Palette Color Mode G	OSDPLTG	0x3004A	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
OSD Palette Color Mode B	OSDPLTB	0x3004C	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
OSD Field control register	OSDFLD	0x3004E	–	–	–	–	–	–	–	–	–	–	x	0	0	1	1	0
OSD Control Register	OSDCON	0x30050	0	0	0	0	–	0	0	0	0	0	0	0	0	0	0	0
OSD FADE Control Register	FADECON	0x30052	–	–	–	–	0	0	0	0	0	0	0	0	0	0	0	0
OSD Background Color Register	OSDBGD	0x30054	0	0	0	0	0	0	0	0	–	0	0	0	0	0	0	0
OSD Margin Control Register	OSDVMGN	0x30056	–	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
OSD Counter Register	OSDCNT	0x30058	–	–	–	–	x	x	x	x	–	–	x	x	x	x	x	x
Locations 0x3005A - 0x3005F are not mapped.																		
IRQ vector register0	IRQV0	0x30060	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register1	IRQV1	0x30062	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register2	IRQV2	0x30064	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register3	IRQV3	0x30066	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6-1. Set 1 Register Values after a Reset (Continued)

Register Name	Mnemonic	Address	Bit Values after a Reset															
		Hex	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ vector register4	IRQV4	0x30068	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register5	IRQV5	0x3006A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register6	IRQV6	0x3006C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register7	IRQV7	0x3006E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register8	IRQV8	0x30070	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register9	IRQV9	0x30072	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register10	IRQV10	0x30074	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register11	IRQV11	0x30076	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register12	IRQV12	0x30078	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register13	IRQV13	0x3007A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register14	IRQV14	0x3007C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRQ vector register15	IRQV15	0x3007E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register0	FIQV0	0x30080	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register1	FIQV1	0x30082	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register2	FIQV2	0x30084	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6-1. Set 1 Register Values after a Reset (Continued)

Register Name	Mnemonic	Address	Bit Values after a Reset															
		Hex	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIQ vector register3	FIQV3	0x30086	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register4	FIQV4	0x30088	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register5	FIQV5	0x3008A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register6	FIQV6	0x3008C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register7	FIQV7	0x3008E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register8	FIQV8	0x30090	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register9	FIQV9	0x30092	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register10	FIQV10	0x30094	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register11	FIQV11	0x30096	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register12	FIQV12	0x30098	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register13	FIQV13	0x3009A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register14	FIQV14	0x3009C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIQ vector register15	FIQV15	0x3009E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Location 0x300A0 - 0x300A7 are not mapped																		
IRQ vector shadow register	IRQSR	0x300A8	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
FIQ vector shadow register	FIQSR	0x300AC	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

☞ **PROGRAM TIP – Initial Program**

```
;*=============================================================================*
;*                    Mode & Stack Define                                      *
;*=============================================================================*
;Pre-defined constants
USERMODE        EQU     0x10
FIQMODE         EQU     0x11
IRQMODE         EQU     0x12
SVCMODE         EQU     0x13
ABORTMODE       EQU     0x17
UNDEFMODE       EQU     0x1b
MODEMASK        EQU     0x1f
NOINT           EQU     0xc0

AbortStack      EQU     0x21c00         ;  20B00–20BFF
UndefStack      EQU     0x21d00         ;  20C00–20CFF
IRQStack        EQU     0x20e00         ;  20D00–20DFF
FIQStack        EQU     0x20f00         ;  20E00–20EFF
SVCStack        EQU     0x21000         ;  20F00–20FFF
;*=============================================================================*
;                   MACRO define                                              *
;*=============================================================================*
                MACRO
                VectorIRQ  $mval1,$mval2    ;  mval1 = IRQ,FIQ register, mval2 = Vector lable
                LDR     r0,=$mval1      ;  r0 = Address of IRQ/FIQ
                LDR     r1,=$mval2      ;  r1 = Address of Service routine for Interrupt
                MOV     r1,r1,lsr #2
                SUB     r1,r1,#8
                STRH    r1,[r0]         ;  Store address
                MEND

                MACRO
                VectorFRQ  $mval1,$mval2    ;  mval1 = IRQ,FIQ register, mval2 = Vector lable
                LDR     r0,=$mval1      ;  r0 = Address of IRQ/FIQ
                LDR     r1,=$mval2      ;  r1 = Address of Service routine for Interrupt
                MOV     r1,r1,lsr #2
                SUB     r1,r1,#9
                STRH    r1,[r0]         ;  Store address
                MEND
```

☞ **PROGRAM TIP – Initial Program (Continued)**

```
;*=============================================================================*
                AREA    init_set,CODE,READONLY
                ENTRY


                b       ResetHandler
                b       HandlerUndef    ;  handlerUndef
                b       HandlerSWI      ;  SWI interrupt handler
                b       HandlerPabort   ;  handlerPAbort
                b       HandlerDabort   ;  handlerDAbort
                b                       ;  handlerReserved
                b       HandlerIRQ
                b       HandlerFIQ

                subs    pc,lr,#4

HandlerFIQ
                                        ;  FIQ service routine in ONE vector (SYSCON.1="0")
                subs    pc,lr,#4        ;  return from HandlerFIQ

HandlerIRQ
                                        ;  IRQ service routine in ONE vector (SYSCON.1="0")
                subs    pc,lr,#4        ;  return from HandlerIRQ

HandlerUndef
                                        ;  Undefined Instruction Exception service
                mov     pc,lr           ;  return from HandlerUndef

HandlerSWI
                                        ;  SWI exception
                mov     pc,lr           ;  return from HandlerSWI

HandlerDabort
                                        ;  Data Abort Exception service
                subs    pc,lr,#8        ;  return from HandlerDabort

HandlerPabort
                                        ;  Instruction Prefetch Abort Exception service
                subs    pc,lr,#4        ;  return from HandlerPabort
```

☞ PROGRAM TIP – Initial Program (Continued)

```
ResetHandler
        ldr     r0,=BTCON       ;  [0] = Clear WT counter
        ldr     r1,=0xA503      ;  [1] = Clear basic counter
        strh    r1,[r0]         ;  [3:2] = Clock source
                                ;  00=Xin/2^4, 01=Xin/2^3, 10=Xin/2^2, 11=Xin
                                ;  (125ms/1s, 62.5/500ms, 31.25/250ms, 7.8/62.5ms)
                                ;  [15:8] Disable watchdog timer (DI=0xA5)

        ldr     r0,=PLLCON      ;  [15–8] = Select Fvco (0x96(40MHz–0xCE(54MHz))
        ldr     r1,=0xB632      ;  Fvco = 48 MHz
        strh    r1,[r0]         ;  [6:4,2:0] = Select OSD/CPU clock (Fvco/1–Fvco/8)
                                ;  CPU = 48/3(16 MHz), OSD 48/7(6.86 MHz)
                                ;  [3] : 0 = Normal operation mode
                                ;  1 = Sleep mode

        ldr     r0,=SYSCON      ;  [0] = Wait enable (2 clock)
        ldr     r1,=0x3         ;  [1] = Enable shadow register
        strh    r1,[r0]         ;  [2] = Enable instruction prefetch

        ldr     sp,=SVCStack    ;  Stack=UserStacks,But Mode=SVCMode
                                ;  Stack initialization is needed using ICE

        mrs     r0,cpsr
        bic     r0,r0,#MODEMASK
        orr     r1,r0,#UNDEFMODE|NOINT
        msr     cpsr,r1         ;  UndefMode
        ldr     sp,=UndefStack

        orr     r1,r0,#ABORTMODE|NOINT
        msr     cpsr,r1         ;  AbortMode
        ldr     sp,=AbortStack

        orr     r1,r0,#IRQMODE|NOINT
        msr     cpsr,r1         ;  IRQMode
        ldr     sp,=IRQStack

        orr     r1,r0,#FIQMODE|NOINT
        msr     cpsr,r1         ;  FIQMode
        ldr     sp,=FIQStack

        bic     r0,r0,#MODEMASK|NOINT
        orr     r1,r0,#SVCMODE
        msr     cpsr,r1         ;  SVCMode
        ldr     sp,=SVCStack
```

☞ **PROGRAM TIP – Initial Program (Continued)**

```
;<< Interrupt Vector address set in Multi Vector System (SYSCON.1="1" >> ==========
                VectorIRQ       IRQV0, Post_Hsync_int    ;
                VectorIRQ       IRQV1, Pre_Hsync_int
                VectorIRQ       IRQV2, Row_int
                VectorIRQ       IRQV3, T0_int
                VectorIRQ       IRQV4, T1_int
                VectorIRQ       IRQV5, T2_int
                VectorIRQ       IRQV6, IR_int
                VectorIRQ       IRQV7, FIFO_int
                VectorIRQ       IRQV8, R_CntOvf_int
                VectorIRQ       IRQV9, V_sync_int
                VectorIRQ       IRQV10,BT_int
                VectorIRQ       IRQV11,P20_int
                VectorIRQ       IRQV12,P21_int
                VectorIRQ       IRQV13,P22_int
                VectorIRQ       IRQV14,P23_int
                VectorIRQ       IRQV15,RT_int

                VectorFRQ       FIQV0, Post_Hsync_int
                VectorFRQ       FIQV1, Pre_Hsync_int
                VectorFRQ       FIQV2, Row_int
                VectorFRQ       FIQV3, T0_int
                VectorFRQ       FIQV4, T1_int
                VectorFRQ       FIQV5, T2_int
                VectorFRQ       FIQV6, IR_int
                VectorFRQ       FIQV7, FIFO_int
                VectorFRQ       FIQV8, R_CntOvf_int
                VectorFRQ       FIQV9, V_sync_int
                VectorFRQ       FIQV10,BT_int
                VectorFRQ       FIQV11,P20_int
                VectorFRQ       FIQV12,P21_int
                VectorFRQ       FIQV13,P22_int
                VectorFRQ       FIQV14,P23_int
                VectorFRQ       FIQV15,RT_int
```

☞ PROGRAM TIP – Initial Program (Continued)

```
;<< Interrupt Setting >> ============================================================

                ldr     r0,=INTMD       ;   0 = IRQ mode (= Reset value)
                mov     r1,=0x7         ;   1 = FIQ mode = Post_Hsync,Pre_Hsync, OSD_row
                                        ;   interrupt
                strh    r1,[r0]

                ldr     r0,=INTPR       ;   0 = Pending clear (= Reset value)
                mov     r1,=0
                strh    r1,[r0]

                ldr     r0,=INTMK       ;   Enable Post_Hsync,Pre_Hsync, OSD_row interrupt
                mov     r1,=0x7
                strh    r1,[r0]

;<< Port setting>> ==================================================================

;<< Timer Setting >> ================================================================

;<< PWM setting>> ===================================================================

;<< OSD setting>> ===================================================================

;<< A/D & Remocon Receive Interrupt setting>> ======================================

;<< Internal RAM & OSD RAM clear >> ================================================;
cold_start
                ldr     r0,=0x20000     ;   Start address of RAM
                ldr     r2,=0x21000
                mov     r1,#0
loop_ram_clear
                str     r1,[r0],#4
                cmp     r0,r2
                blt     loop_ram_clear

main_start

                                        ;   main service program
                b       start_main
```

☞ PROGRAM TIP – Initial Program (Continued)

```
;<< Interrupt service Program >>
Post_Hsync_int
Pre_Hsync_int
Row_int
T0_int
T1_int
T2_int
IR_int
FIFO_int
R_CntOvf_int
V_sync_int
BT_int
P20_int
P21_int
P22_int
P23_int
RT_int
                    subs        pc,lr,#4
```

POWER-DOWN MODES

SLEEP MODE

Sleep mode is invoked by setting PLLCON register bit 3 to high. In Sleep mode, the operation of the CPU and all peripherals is halted. At this time the CPU clock stops while the on-chip main oscillator keeps operating. The supply current is reduced to less than 2 mA. All system functions stop when the clock "freezes," but data stored in the internal register file is retained. Sleep mode can be released in one of two ways: by a reset or by an interrupt.

Using RESET to Release Sleep Mode

Sleep mode is released when the RESET signal returns from low to high level: all system and peripheral control registers are reset to their default hardware values and the contents of all data registers are retained. A reset operation automatically selects a slow clock (1/16) because BTCON.3, and BTCON.2 are cleared to '00B'. After the programmed oscillation stabilization interval has elapsed, the CPU starts the system initialization routine by fetching the program instruction stored in location 0x00.

In summary, the following events occur when SLEEP mode is released by RESET:

1. A power-on reset releases SLEEP mode and PLL block starts operation. But the Oscillator Does NOT stop in SLEEP mode.
2. When a reset occurs in SLEEP mode, the basic timer counter increases at the rate of $X_{IN}/2^4$.
3. Clock oscillation stabilization interval begins and continues until bit2 (WTCNT.2) of 3-bit Watchdog timer (WTCNT) is set (that is 500ms).
4. When bit 2 of 3-bit Watchdog timer is set, a normal CPU operation resumes.

Using Interrupt to Release Sleep Mode

In summary, the following events occur when SLEEP mode is released by INTERRUPT:

1. External interrupt 0, 1, 2, 3, V-sync interrupt, remocon signal input interrupt and real timer interrupt occurs to release the SLEEP mode release and PLL block starts. But the Oscillator Does NOT stop in SLEEP mode.
2. When SLEEP mode is released , the BTCNT value increases at the rate of the $X_{IN}/1$.
3. Clock oscillation stabilization interval begins and continues until bit 7 (BTCNT.7) of 8-bit Basic timer is set (that is 3.90625ms).
4. When bit7 (BTCNT.7) of 8-bit Basic timer is set, a normal CPU operation resumes

SYSTEM CONTROL REGISTER

CLOCK WAIT FUNCTION

SYSCON.0 controls ROM access time. When SYSCON.0 is set to “0”, a read/write operation is performed in 2 clock, and when it is set to “1” the operation is done in 1 clock.

Interrupt Type Selection

SYSCON.1 is used to select an interrupt type: either one-vector type (SYSCON.1 = “0”) or multi-vector type with shadow register (SYSCON.1 = “1”).

Instruction Prefetch Function

SYSCON.2 enables instruction fetch in 32-bit unit when a thumb code, which is in 16-bit, is used. When it is set to “1”, a 32-bit instruction code is fetched from ROM. The upper 16 bits of the instruction code is temporarily stored in the buffer, and fetched without additional order. This function can substantially reduce the time for instruction fetch by allowing two 16-bit thumb instruction codes to be fetched from ROM in one operation. This is effecting only when the clock wait function is enabled (SYSCON.0 = “0”). When SYSCON.0 = “1”, the function is not available.

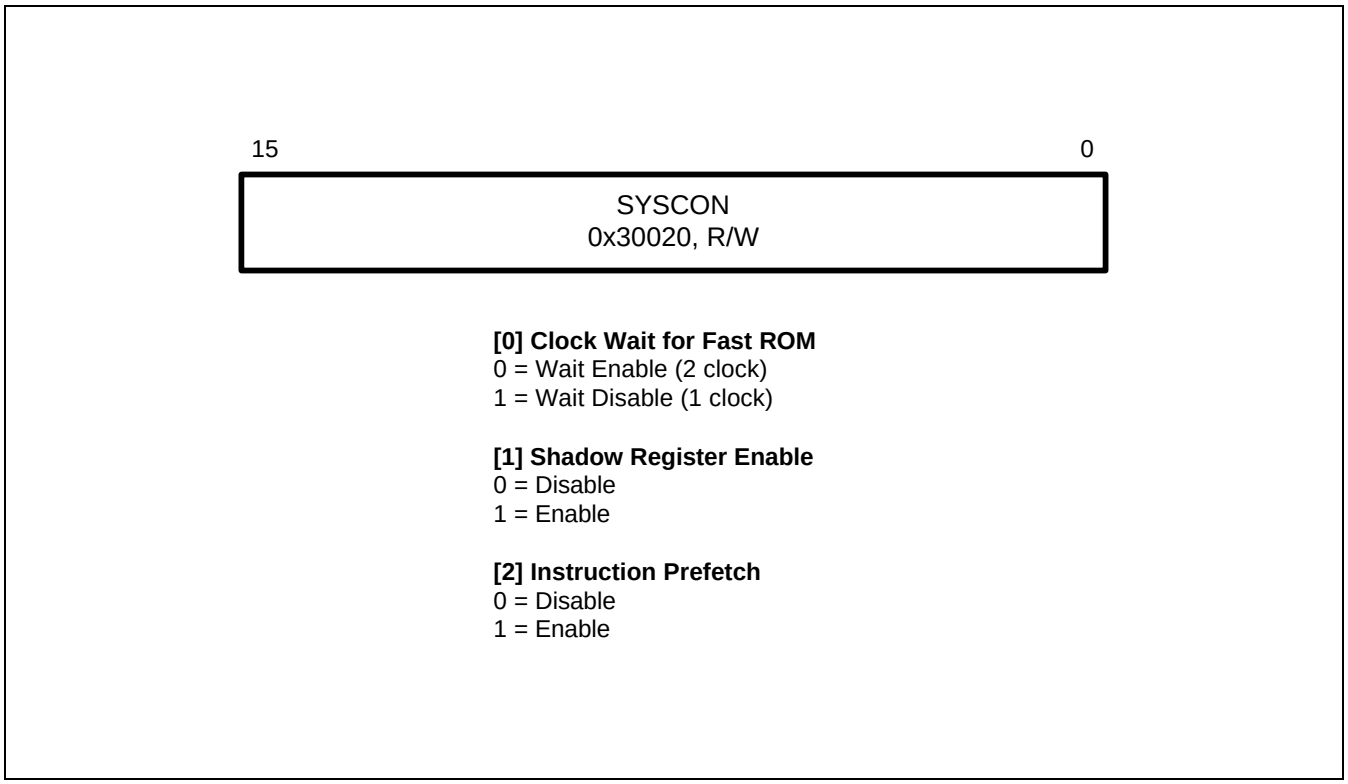


Figure 6-3. System Control Register

7 CLOCK CIRCUITS

OVERVIEW

The clock frequency generated by an external crystal, 32,768 Hz. The maximum CPU clock frequency is 16 MHz. The X_{IN} and X_{OUT} pins connect the external oscillation source to the on-chip clock circuit.

The LPF pin connects the LPF to the external circuit for the PLL operation (refer to diagram below). The S3C380D divides the PLL circuit-generated F_{VCO} into 2, 3, 4, 5, 6, 7, or 8 using the CPU clock divider PLLCON.2-0, to use as the CPU clock (f_{CPU}). It also divides the f_{VCO} into 2, 3, 4, 5, 6, 7, or 8 using the OSD clock divider PLLCON.6-4, to use as the OSD clock (f_{OSD}). The F_{VCO} can generate a clock frequency between 40 MHz and 54 MHz, in units of 0.25 MHz, using the PLLCON.15-8.

SYSTEM CLOCK CIRCUIT

The system clock circuit has the following components:

— External crystal or ceramic oscillation source, 32,768 Hz
— Main system clock, f_{VCO}, stop and wake-up functions
— Programmable frequency divider for the main system clock (40MHz - 54 MHz)
— Programmable frequency divider for the CPU clock (f_{VCO} divided by 2, 3, 4, 5, 6, 7, or 8)
— Programmable frequency divider for the OSD clock (f_{VCO} divided by 2, 3, 4, 5, 6, 7, or 8)
— Clock (PLL) circuit control register, PLLCON

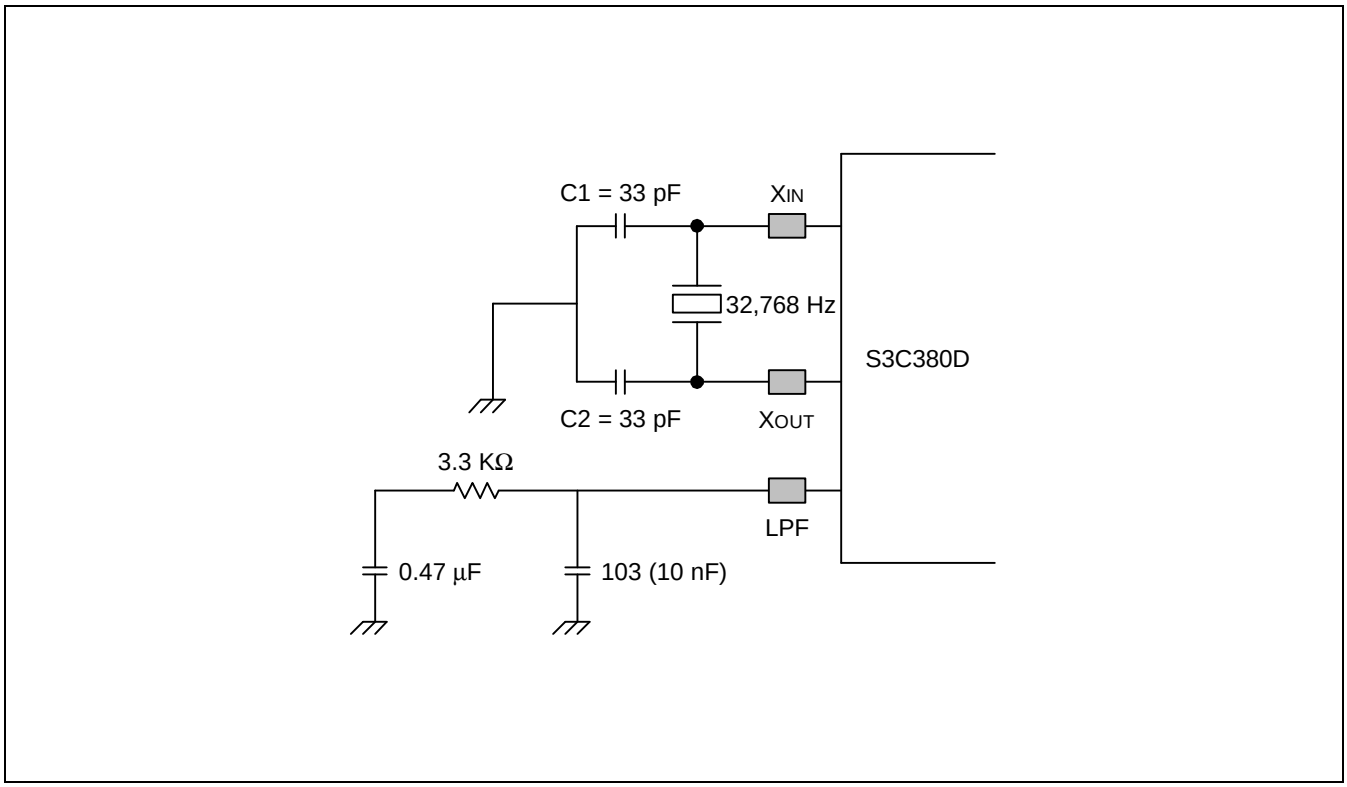


Figure 7-1. Main Oscillator Circuit (External Crystal or Ceramic Resonator)

CLOCK STATUS DURING POWER-DOWN MODES

The power-down mode, Sleep mode, affects system clock oscillation as follows:

— In Sleep mode, the main system clock (f_{VCO}) is halted while external oscillation is not. When sleep mode is released, by a reset operation or by an external interrupt, the PLL operation (main system clock) starts.

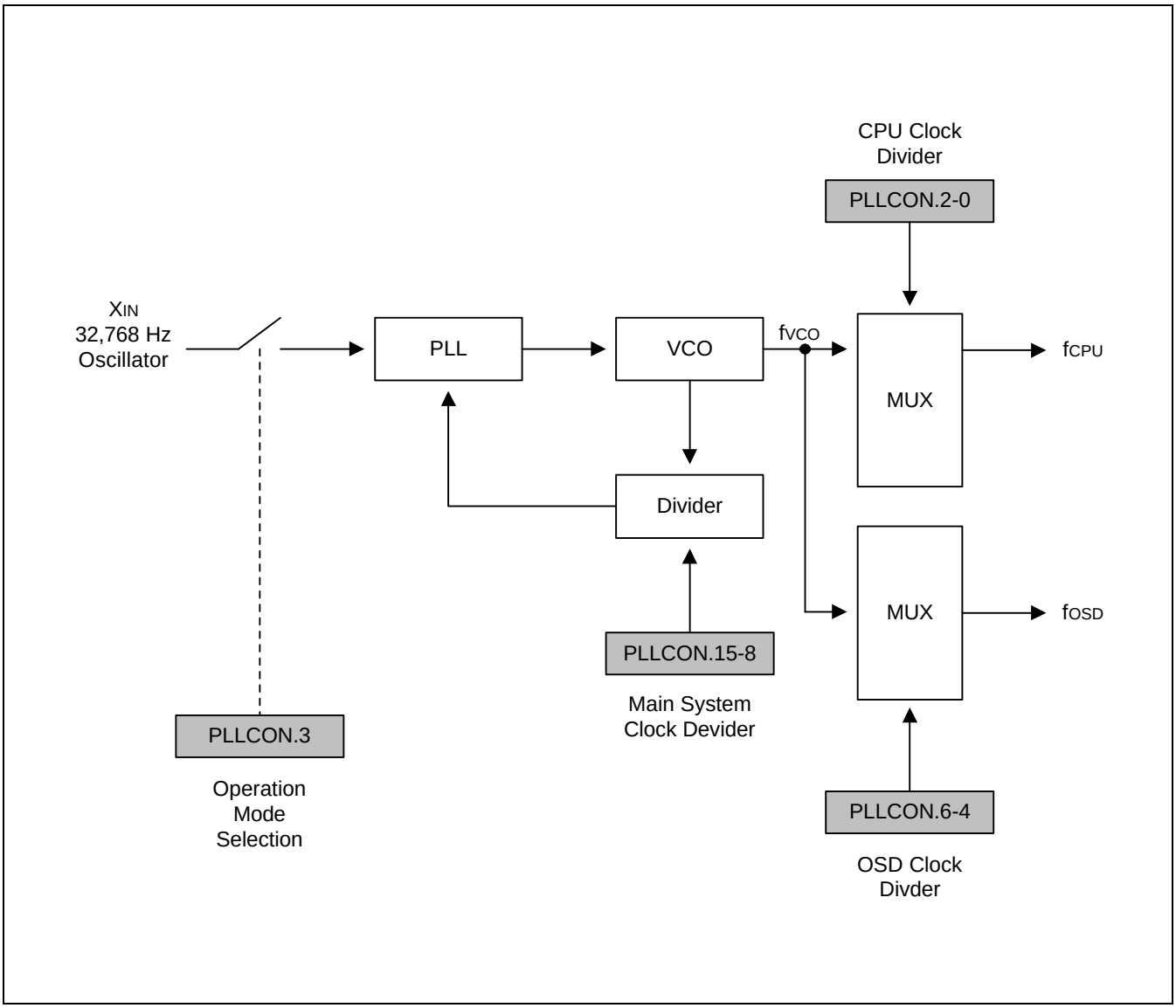


Figure 7-2. PLL Block Diagram

SYSTEM CLOCK CONTROL REGISTER (PLLCON)

The system clock control register, PLLCON, is located at address 0x3001E. It is read/write addressable and has the following functions:

— Main system clock, f_{VCO}, stop control

— Main system clock, f_{VCO}, frequency set value: f_{VCO} = 40 MHz-54 MHz

— CPU clock signal selection (f_{CPU}); f_{VCO} divided by 2, 3, 4, 5, 6, 7, or 8

— OSD clock signal selection (f_{OSD}); f_{VCO} divided by 2, 3, 4, 5, 6, 7, or 8

Registers	Address	R/W	Description	Reset Value
PLLCON	0x3001E	R/W	PLL clock control register	B763h

[2:0] fCPU
000 = Not used
001 = fvco/2
010 = fvco/3
011 = fvco/4
100 = fvco/5
101 = fvco/6
110 = fvco/7
111 = fvco/8

[3] Operation mode
0 = Normal Operation Mode
1 = SLEEP Mode

[6:4] fOSD
000 = Not used
001 = fvco/2
010 = fvco/3
011 = fvco/4
100 = fvco/5
101 = fvco/6
110 = fvco/7
111 = fvco/8

[7] Not used

[15:8]

Binary	HEX	fVCO
1001 0110	(96H)	= 40.00 MHz
1001 0111	(97H)	= 40.25 MHz
1001 1000	(98H)	= 40.50 MHz
1001 1001	(99H)	= 40.75 MHz
1001 1010	(9AH)	= 41.00 MHz
1001 1011	(9BH)	= 41.25 MHz
⋮	⋮	⋮
1100 1001	(C9H)	= 52.75 MHz
1100 1010	(CAH)	= 53.00 MHz
1100 1011	(CBH)	= 53.25 MHz
1100 1100	(CCH)	= 53.50 MHz
1100 1101	(CDH)	= 53.75 MHz
1100 1110	(CEH)	= 54.00 MHz

NOTE: fVCO Default Value = B7H (48.25 MHz)
fCPU Default Value = 3H (12.06 MHz)
fOSD Default Value = 6H (6.89 MHz)

Figure 7-3. System Clock Control Register (PLLCON)

Table 7-1. f_{CPU} and f_{OSD} Clock Value by PLLCON Control

PLLCON.15-8 (HEX)	f_{VCO} (MHz)	$f_{VCO}/2$	$f_{VCO}/3$	$f_{VCO}/4$	$f_{VCO}/5$	$f_{VCO}/6$	$f_{VCO}/7$	$f_{VCO}/8$
96h	40.00	20.00	13.33	10.00	8.00	6.67	5.71	5.00
97h	40.25	20.13	13.42	10.06	8.05	6.71	5.75	5.03
98h	40.50	20.25	13.50	10.13	8.10	6.75	5.79	5.06
99h	40.75	20.38	13.58	10.19	8.15	6.79	5.82	5.09
9Ah	41.00	20.50	13.67	10.25	8.20	6.83	5.86	5.13
9Bh	41.25	20.63	13.75	10.31	8.25	6.88	5.89	5.16
9Ch	41.50	20.75	13.83	10.38	8.30	6.92	5.93	5.19
9Dh	41.75	20.88	13.92	10.44	8.35	6.96	5.96	5.22
9Eh	42.00	21.00	14.00	10.50	8.40	7.00	6.00	5.25
9Fh	42.25	21.13	14.08	10.56	8.45	7.04	6.04	5.28
A0h	42.50	21.25	14.17	10.63	8.50	7.08	6.07	5.31
A1h	42.75	21.38	14.25	10.69	8.55	7.13	6.11	5.34
A2h	43.00	21.50	14.33	10.75	8.60	7.17	6.14	5.38
A3h	43.25	21.63	14.42	10.81	8.65	7.21	6.18	5.41
A4h	43.50	21.75	14.50	10.88	8.70	7.25	6.21	5.44
A5h	43.75	21.88	14.58	10.94	8.75	7.29	6.25	5.47
A6h	44.00	22.00	14.67	11.00	8.80	7.33	6.29	5.50
A7h	44.25	22.13	14.75	11.06	8.85	7.38	6.32	5.53
A8h	44.50	22.25	14.83	11.13	8.90	7.42	6.36	5.56
A9h	44.75	22.38	14.92	11.19	8.95	7.46	6.39	5.59
AAh	45.00	22.50	15.00	11.25	9.00	7.50	6.43	5.63
ABh	45.25	22.63	15.08	11.31	9.05	7.54	6.46	5.66
ACh	45.50	22.75	15.17	11.38	9.10	7.58	6.50	5.69
ADh	45.75	22.88	15.25	11.44	9.15	7.63	6.54	5.72
AEh	46.00	23.00	15.33	11.50	9.20	7.67	6.57	5.75
AFh	46.25	23.13	15.42	11.56	9.25	7.71	6.61	5.78

Table 7-1. f_{CPU} and f_{OSD} Clock Value by PLLCON Control (Continued)

PLLCON.15-8 (HEX)	f_{VCO} (MHz)	$f_{VCO}/2$	$f_{VCO}/3$	$f_{VCO}/4$	$f_{VCO}/5$	$f_{VCO}/6$	$f_{VCO}/7$	$f_{VCO}/8$
B0h	46.50	23.25	15.50	11.63	9.30	7.75	6.64	5.81
B1h	46.75	23.38	15.58	11.69	9.35	7.79	6.68	5.84
B2h	47.00	23.50	15.67	11.75	9.40	7.83	6.71	5.88
B3h	47.25	23.63	15.75	11.81	9.45	7.88	6.75	5.91
B4h	47.50	23.75	15.83	11.88	9.50	7.92	6.79	5.94
B5h	47.75	23.88	15.92	11.94	9.55	7.96	6.82	5.97
B6h	48.00	24.00	16.00	12.00	9.60	8.00	6.86	6.00
B7h	48.25	24.13	16.08	12.06	9.65	8.04	6.89	6.03
B8h	48.50	24.25	16.17	12.13	9.70	8.08	6.93	6.06
B9h	48.75	24.38	16.25	12.19	9.75	8.13	6.96	6.09
BAh	49.00	24.50	16.33	12.25	9.80	8.17	7.00	6.13
BBh	49.25	24.63	16.42	12.31	9.85	8.21	7.04	6.16
BCh	49.50	24.75	16.50	12.38	9.90	8.25	7.07	6.19
BDh	49.75	24.88	16.58	12.44	9.95	8.29	7.11	6.22
BEh	50.00	25.00	16.67	12.50	10.00	8.33	7.14	6.25
BFh	50.25	25.13	16.75	12.56	10.05	8.38	7.18	6.28
C0h	50.50	25.25	16.83	12.63	10.10	8.42	7.21	6.31
C1h	50.75	25.38	16.92	12.69	10.15	8.46	7.25	6.34
C2h	51.00	25.50	17.00	12.75	10.20	8.50	7.29	6.38
C3h	51.25	25.63	17.08	12.81	10.25	8.54	7.32	6.41
C4h	51.50	25.75	17.17	12.88	10.30	8.58	7.36	6.44
C5h	51.75	25.88	17.25	12.94	10.35	8.63	7.39	6.47
C6h	52.00	26.00	17.33	13.00	10.40	8.67	7.43	6.50
C7h	52.25	26.13	17.42	13.06	10.45	8.71	7.46	6.53
C8h	52.50	26.25	17.50	13.13	10.50	8.75	7.50	6.56
C9h	52.75	26.38	17.58	13.19	10.55	8.79	7.54	6.59
CAh	53.00	26.50	17.67	13.25	10.60	8.83	7.57	6.63
CBh	53.25	26.63	17.75	13.31	10.65	8.88	7.61	6.66
CCh	53.50	26.75	17.83	13.38	10.70	8.92	7.64	6.69
CDh	53.75	26.88	17.92	13.44	10.75	8.96	7.68	6.72
CEh	54.00	27.00	18.00	13.50	10.80	9.00	7.71	6.75

8 INTERRUPT

OVERVIEW

The S3C380D/F380D has two interrupt levels, IRQ and FIQ, each of which has 16 interrupt sources. An interrupt request can be generated at an internal function block external pins. As ARM7TDMI core recognizes two kinds of interrupt level: a normal interrupt request (IRQ), and a fast interrupt request (FIQ), all the S3C380D/F380D interrupts are categorized as either IRQ or FIQ. Each providing 16 interrupt sources, these levels are controlled by the interrupt special register which is different according to the method of interrupt service: one-vector method or multi-vector method.

In the one vector method, only vector addresses corresponding to each level exist regardless of interrupt sources. An IRQ interrupt request uses 0x18 as its vector address while an FIQ interrupt request uses 0x1C. In the multi-vector method, all the 16 IRQ, and 16 FIQ interrupt sources can designate their own interrupt vectors. In the S3C380D/F380D interrupt controller, the one vector method is controlled by the special registers of interrupt mode register (INTMD), and the multi-vector method is by those of FIQ vector shadow register (FIQSR).

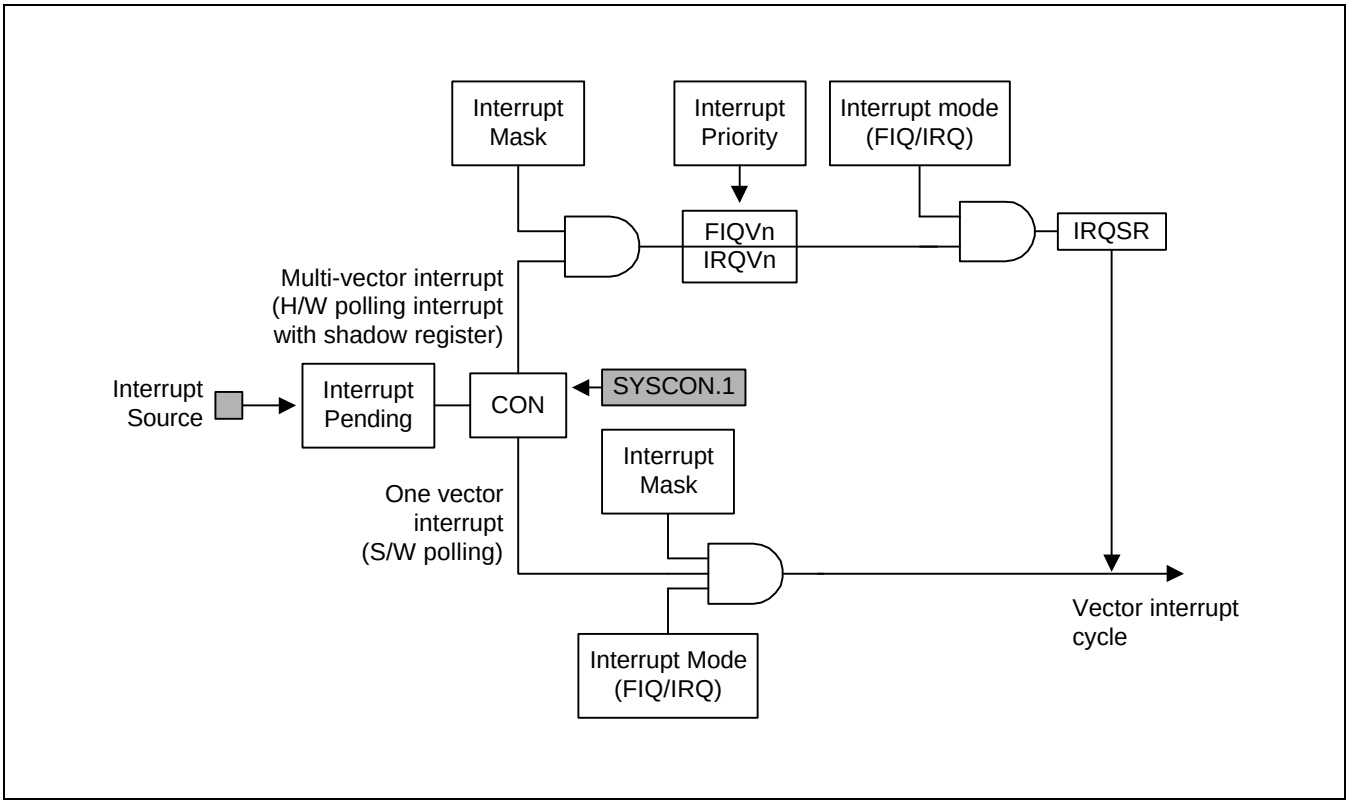


Figure 8-1. S3C380D Interrupt Structure

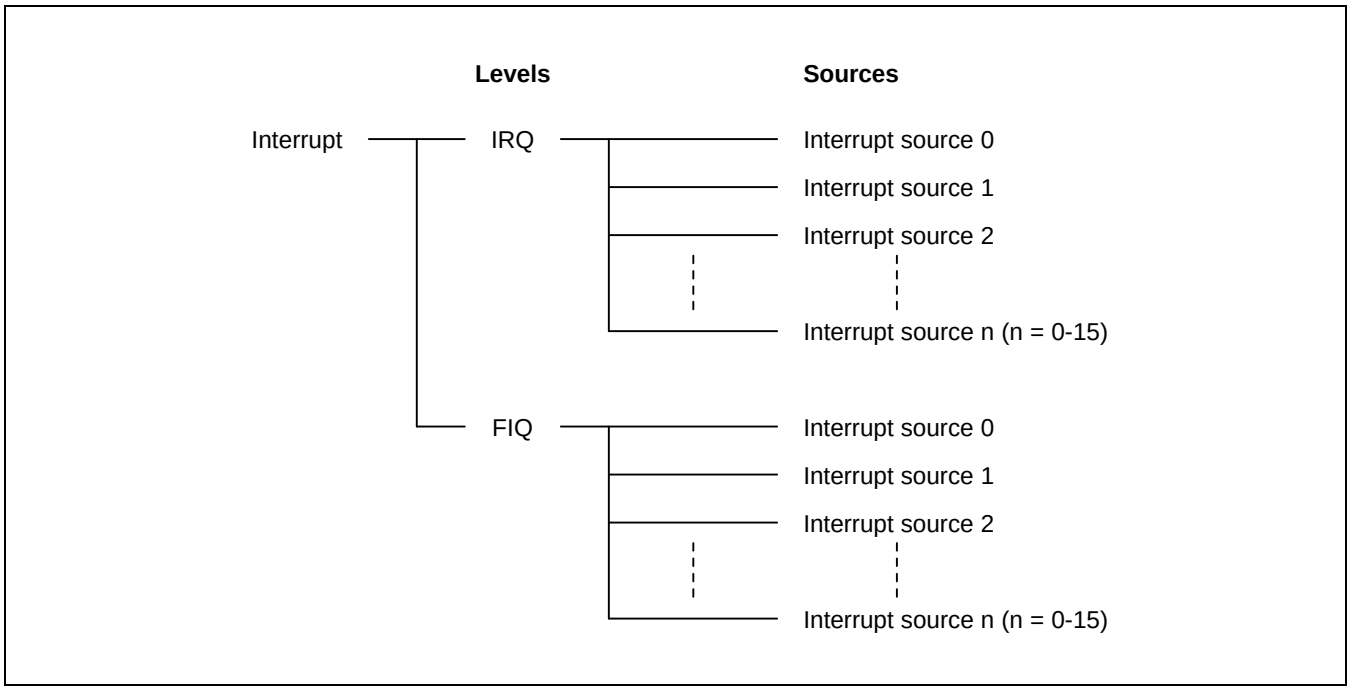


Figure 8-2. Interrupt Hierachy

Table 8-1. Interrupt Source Description

Interrupt Number	Interrupt Source	Description	Default Priority
0	Post-H-sync interrupt	Every H-sync input end edge; falling/rising	1
1	Pre-H-sync interrupt	Every H-sync input start edge; falling/rising	1
2	OSD row interrupt	OSD row start point (line 0 of every row)	2
3	Timer 0 interrupt	16-bit timer 0 counter overflow	3
4	Timer 1 interrupt	16-bit timer 1 counter overflow	4
5	Timer 2 interrupt	16-bit timer 2 counter overflow	5
6	Romote control signal input interrupt (capture)	Signal input for capture at IRIN (P36); remote control	6
7	FIFO full interrupt	8 FIFO full in Remocon receiver block	7
8	Remocon counter overflow interrupt	Counter overflow in Remocon receiver block	8
9	V-sync interrupt	V-sync input edge; falling/rising	9
10	Basic timer interrupt	Basic timer counter overflow	10
11	External interrupt 0	External input at INT0 (P14); falling/rising	11
12	External interrupt 1	External input at INT1 (P15); falling/rising	12
13	External interrupt 2	External input at INT2 (P16); falling/rising	13
14	External interrupt 3	External input at INT3 (P17); falling/rising	14
15	Real time interrupt	Real time 1 sec interrupt	15

INTERRUPT SOURCES

The 16 interrupt sources in the S3C380D interrupt structure are described, in brief, as follows:

[15] Real timer interrupt
[14] External interrupt 3
[13] External interrupt 2
[12] External interrupt 1
[11] External interrupt 0
[10] Basic timer interrupt
[9] V-sync interrupt
[8] Remocon counter overflow interrupt
[7] FIFO full interrupt
[6] Remote control signal input interrupt (capture)
[5] Timer 2 interrupt
[4] Timer 1 interrupt
[3] Timer 0 interrupt
[2] OSD row interrupt
[1] Pre-H-sync interrupt (1st edge H-sync)
[0] Post-H-sync interrupt (2nd edge H-sync)

INTERRUPT CONTROLLER SPECIAL REGISTERS

INTERRUPT MODE REGISTER (INTMD)

The interrupt mode registers specify whether an interrupt is a fast or a normal interrupt mode. Each of 16 bits in the interrupt mode enable register corresponds to the interrupt sources. When the source's interrupt mode bit is set to"1", the interrupt is processed by the ARM7TDMI core in FIQ (fast interrupt) mode. Otherwise, it is processed in IRQ mode (normal interrupt). The 16 interrupt sources are summarized as follows:

Register	Offset Address	R/W	Description	Reset Value
INTMD	0x30024	R/W	Interrupt mode register	0000h

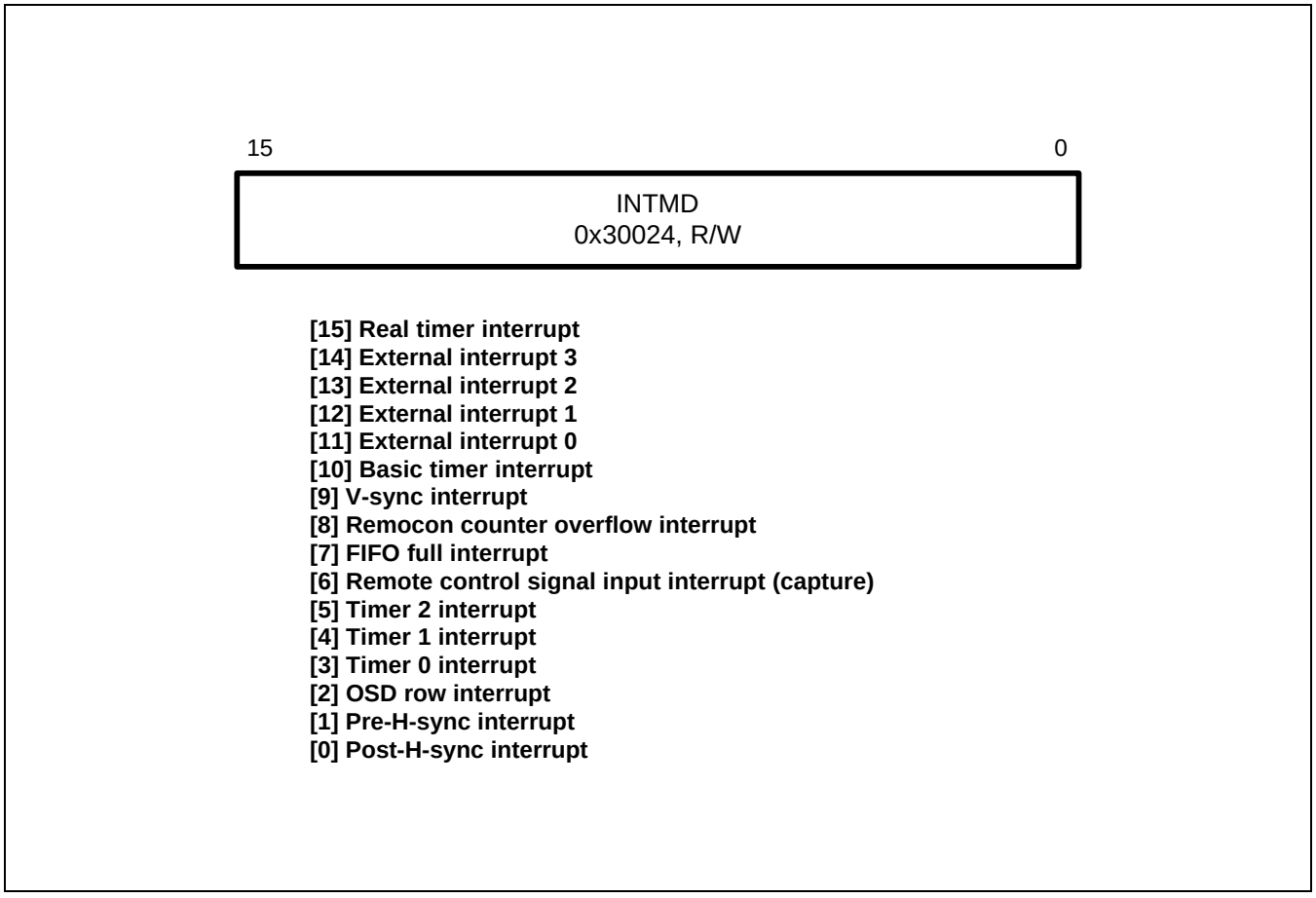


Figure 8-3. Interrupt Mode Register (INTMD)

INTERRUPT PENDING REGISTER (INTPR)

The interrupt pending register, INTPR, contains interrupt pending bits for each interrupt source. Each of the 16 bits in the interrupt pending register corresponds to the interrupt sources. When an interrupt request is generated, it is set to "1". The service routine must clear the appropriate pending bit. To clear a pending bit, write "0" to it. If this bit is "1", it has no effect. The 16 interrupt sources are summarized as follows:

Register	Offset Address	R/W	Description	Reset Value
INTPR	0x30022	R/W	Interrupt pending register	0000h

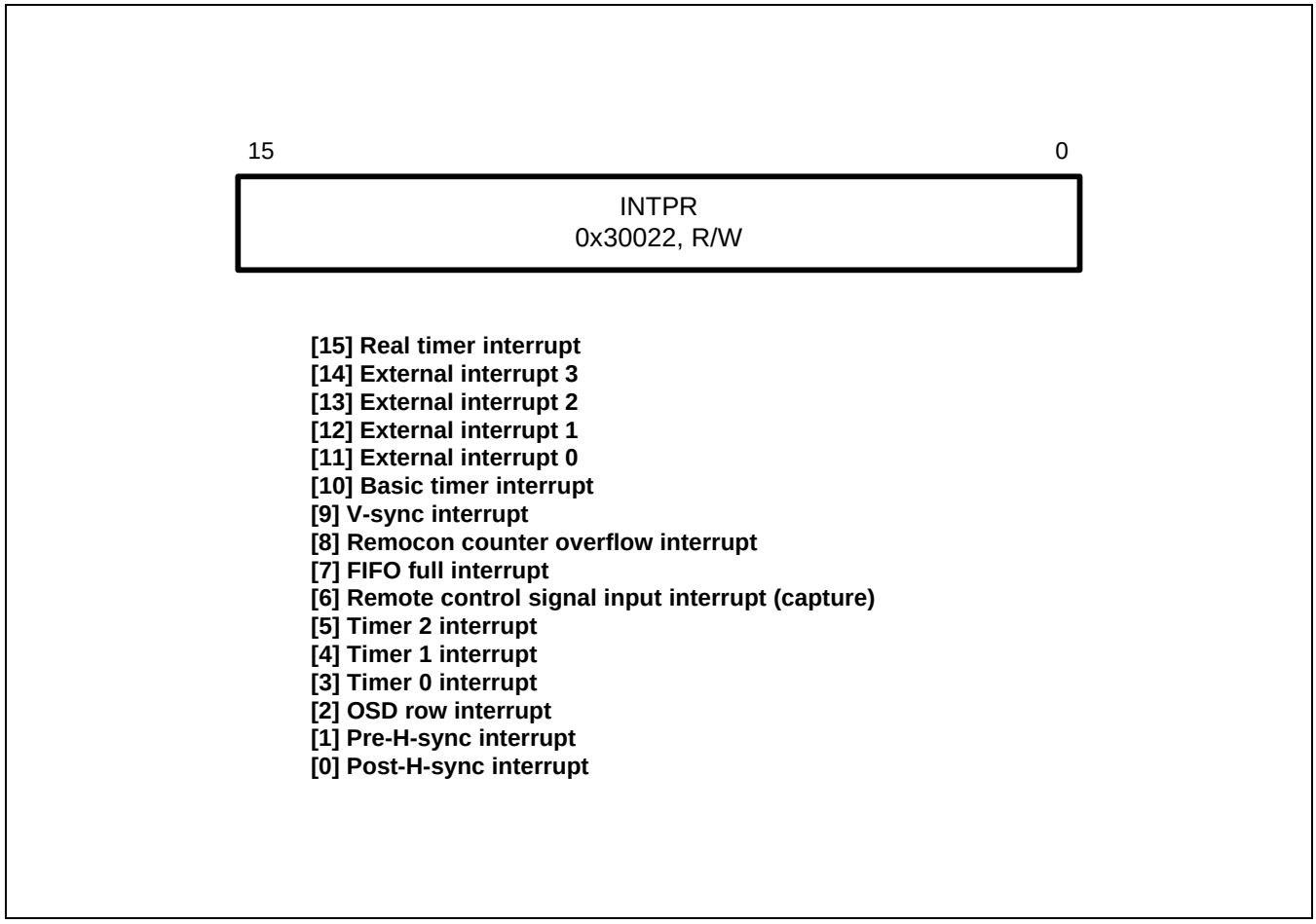


Figure 8-4. Interrupt Pending Register (INTPR)

NOTE

When you clear a pending bit, we recommend you should. “MOV”, “LDR” instruction. If you use other instruction, other pending bit can be cleared.

INTERRUPT MASK REGISTER (INTMK)

The interrupt mask register, INTMK, contains interrupt mask bits for each interrupt source. Each of the 16 bits in the interrupt mask register corresponds to the interrupt sources. If a source's interrupt mask bit is "0", the interrupt is not serviced by CPU when the corresponding interrupt request is generated. If the mask bit is '1', the interrupt is serviced upon request. The 16 interrupt sources are summarized as follows:

Register	Offset Address	R/W	Description	Reset Value
INTMK	0x30026	R/W	Interrupt mask register	0000h

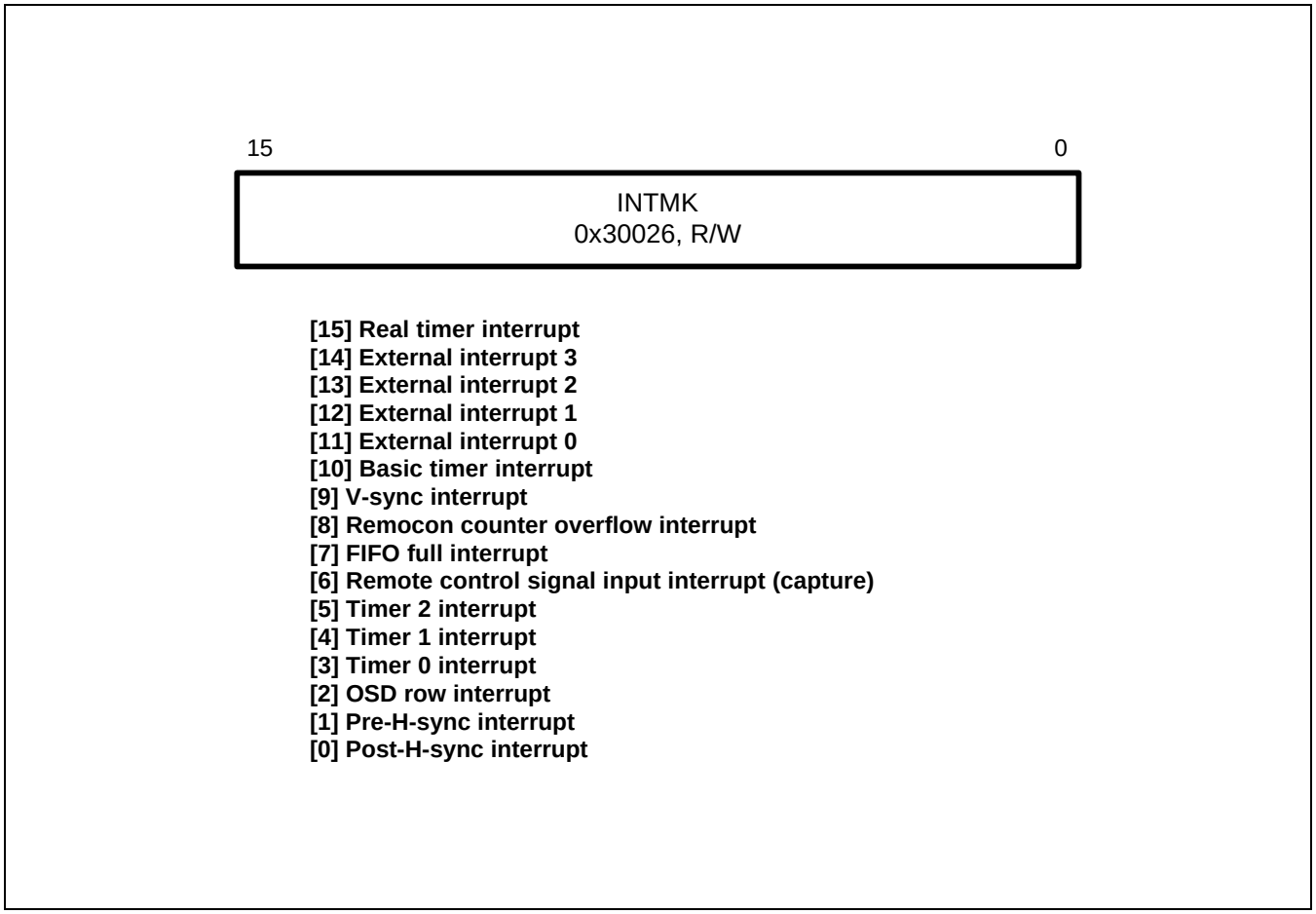


Figure 8-5. Interrupt Mask Register (INTMK)

INTERRUPT PRIORITY REGISTERS (IPR0-IPR3)

Interrupt priority registers enable users to define priority when move than one interrupt occur. Interrupt priority is set by giving the value corresponding to any one of 16 interrupts to interrupt register, IPR0-IPR3.

The value “0001B” in IPR0.0-3 puts pre-H-sync interrupt on the top priority (priority 0). Writing “0111B” in IPR.4-7 selects FIFO full interrupt as the sixth highest priority (priority 5).

Register	Offset Address	R/W	Description	Reset Value
IPR0	0x30028	R/W	Interrupt priority register 0	3210h
IPR1	0x3002A	R/W	Interrupt priority register 1	7654h
IPR2	0x3002C	R/W	Interrupt priority register 2	0BA98h
IPR3	0x3002E	R/W	Interrupt priority register 3	0FEDCh

Register	Bits	Setting	Description
IPR0	3-0	xxxx	Priority 0 setting
	7-4	xxxx	Priority 1 setting
	b-8	xxxx	Priority 2 setting
	f-c	xxxx	Priority 3 setting
IPR1	3-0	xxxx	Priority 4 setting
	7-4	xxxx	Priority 5 setting
	b-8	xxxx	Priority 6 setting
	f-c	xxxx	Priority 7 setting
IPR2	3-0	xxxx	Priority 8 setting
	7-4	xxxx	Priority 9 setting
	b-8	xxxx	Priority 10 setting
	f-c	xxxx	Priority 11 setting
IPR3	3-0	xxxx	Priority 12 setting
	7-4	xxxx	Priority 13 setting
	b-8	xxxx	Priority 14 setting
	f-c	xxxx	Priority 15 setting

IRQ VECTOR ADDRESS REGISTERS (IRQVn, n = 0-15)

IRQ vector address registers serve as a place to store start address offsets of an interrupt service routine generated when multi-vector interrupt is selected (SYSCON.1 = “1”). The registers from IRQV to IRQV15 are saving the start address offset of post-H-sync interrupt (IRQV0), pre-H-sync interrupt (IRQV1), OSD row interrupt (IRQV2), Timer 0 interrupt (IRQV3), Timer 1 interrupt (IRQV4), Timer 2 interrupt (IRQV5), remote control signal input interrupt (capture)(IRQV6), FIFO full interrupt (IRQV7), remocon counter overflow interrupt (IRQV8), V-sync interrupt (IRQV9), basic timer interrupt (IRQV10), external interrupt 0 (IRQV11), external interrupt 1 (IRQV12), external interrupt 2 (IRQV13), external interrupt 3 (IRQV14), and real time interrupt (IRQV15), respectively.

In one vector interrupt (Software polling method), vectors in IRQ vector address registers (IRQVn, n = 1-15) have no meaning as those registers are not employed. Upon receiving an interrupt request in multi-vector interrupt, offsets stored in IRQ vector address registers are automatically loaded to bit 0 to bit 15 of IRQ vector shadow registers. The content of IRQ vector shadow registers comprises the instruction jumping to the start address of the correct interrupt service routine. Interrupt service is performed by executing this instruction.

How to store offsets in IRQ vector address registers. Assuming the service start address is 0x2014, shift 0x2014 by 2 bits to the right. Discard the lower 2 bits, as they are word addresses. Shift 0x18 (IRQ vector address) by 2 bits to the right and subtract it from the value above. Save the resulting value in IRQVn.

Register	Offset Address	R/W	Description	Reset Value
IRQV0-IRQV15	From 0x30060 to 0x3007F	R/W	IRQ Interrupt vector register 0-15	0000h

FIQ VECTOR ADDRESS REGISTERS (FIQVn, n = 0-15)

FIQ vector address registers serve as a place to store start address offsets of an interrupt service routine generated when multi-vector interrupt is selected (SYSCON.1 = “1”). The registers from FIQV0 to FIQV15 are saving the start address offset of post-H-sync interrupt (FIQV0), pre-H-sync interrupt (FIQV1), OSD row interrupt (FIQV2), timer 0 interrupt (FIQV3), timer 1 interrupt (FIQV4), timer 2 interrupt (FIQV5), remote control signal input interrupt (capture) (FIQV6), FIFO full interrupt (FIQV7), remocon counter overflow interrupt (FIQV8), V-sync interrupt (FIQV9), basic timer interrupt (FIQV10), external interrupt 0 (FIQV11), external interrupt 1 (FIQV12), external interrupt 2 (FIQV13), external interrupt 3 (FIQV14), and real time interrupt (FIQV15), respectively.

Register	Offset Address	R/W	Description	Reset Value
FIQV0-FIQV15	From 0x30080 to 0x3009F	R/W	FIQ Interrupt vector register 0-15	0000h

IRQ VECTOR SHADOW REGISTER (IRQSR)

IRQ vector shadow register contains the instruction that sumps to the start address of an interrupt service routine in multi-vector interrupt (SYSCON.1 = “1”). In one vector interrupt (SYSCON.1 = “0”), the IRQ interrupt vector address is 0x18, and in multi-vector interrupt (SYSCON.1 = “1”) it is 0x300A8 (IRQSR).

IRQSR register value configuration

Register	Offset Address	R/W	Description	Reset Value
IRQSR	0x300A8	R	IRQ Interrupt vector shadow register	0000h

FIQ VECTOR SHADOW REGISTER (FIQSR)

FIQ vector shadow register contains the instruction that jumps to the start address of an interrupt service routine in multi-vector interrupt (SYSCON.1 = “1”). In one vector interrupt (SYSCON.1 = “0”), the FIQ interrupt vector address is 0x1C, and in multi-vector interrupt (SYSCON.1 = “1”), it is 0x300AC (IRQSR).

FIQSR register value configuration

Register	Offset Address	R/W	Description	Reset Value
FIQSR	0x300AC	R	FIQ Interrupt vector selection register	0000h

When an interrupt is generated, the vector address offset interrupt contained in the vector address register is loaded to the IRQSR, or the FIQSR register, and the interrupt service jumps to the vector address of IRQSR or FIQSR.

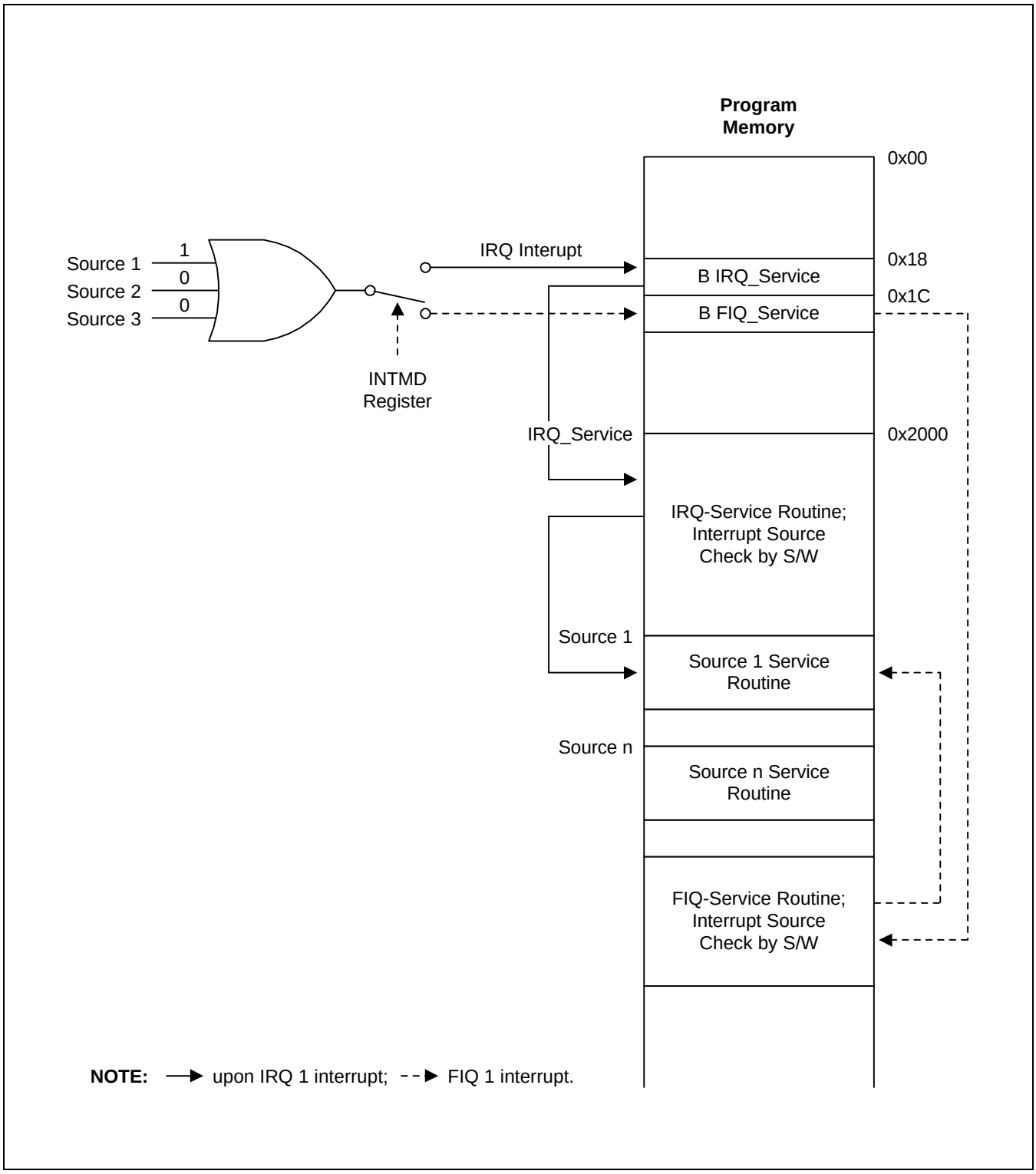


Figure 8-6. One Vector Interrupt Service

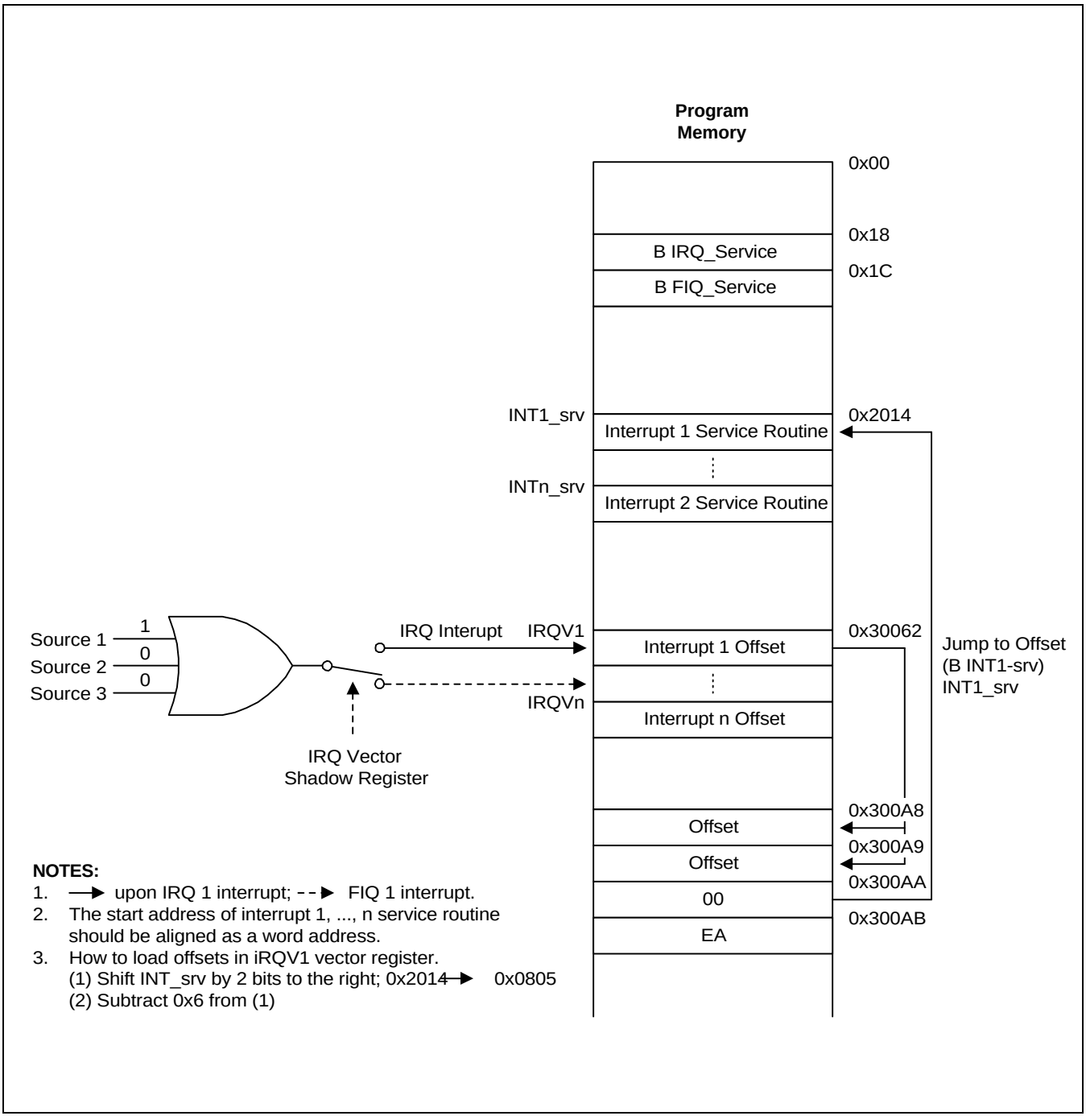


Figure 8-7. Multi-vector Interrupt Service

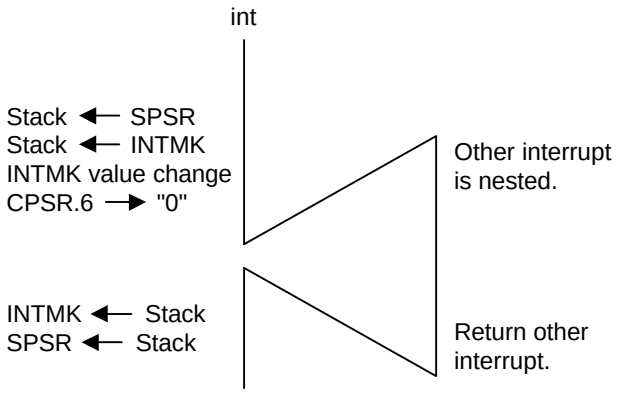
☞ PROGRAM TIP – Interrupt Nesting

```
P20_int
                .
                .
                .

                ldr         r2,=INTPR
                mvn         r0,#0x800           ;   p2.0 pending clear
                strh        r0,[r2],#4          ;   pending clear
                                                ;   r3 ← INTMK address
                mrs         r0,SPSR             ;   r0 ← SPSR

                ldrh        r1,[r2]             ;   r2 = INTMK
                stmfd       sp!,{r0,r1}         ;   push to stack INTMK,SPSR
                mov         r1,#0x1000          ;   only enable P2.1 int
                strh        r1,[r2]
;----- fiq int nesting set -------------;
                mrs         r0,CPSR             ;
                bic         r0,r0,#0x40         ;   fiq receive
                msr         CPSR,r0             ;
                .
                .
                ; < Interrupt is nested >
                .
                ldmfd       sp!,{r0,r1}         ;   pop INTMK
                ldr         r2,=INTMK
                strh        r1,[r2]
                msr         SPSR,r0             ;

                subs        pc,lr,#4            ;   return from P20 interrupt
```



9 I/O PORTS

OVERVIEW

Of the 42 pins in the S3C380D/F380D's SDIP package, 25 pins are used for I/O. There are four ports:

— Three 8-bit I/O ports (port 0, 1, 2)

— One 1-bit I/O port (port 3)

Each port can be easily configured by software to meet various system configuration and design requirements. As the CPU accesses I/O ports by directly writing or reading port register addresses, no special I/O instruction is needed.

Table 9-1. S3C380D Port Configuration Overview

Port		Pin Type	Pin Configuration	Circuit Type	Programmability
P0	P0.0	I/O	Input mode or push-pull output mode is software configurable. P0.0: PWM0 (14-bit PWM Output)	6	Bit programmable
	P0.1-P0.2 P0.3		General I/O Port (3-bit), Input or n-channel open-drain output is software configurable. Pins can withstand up to 6-volt loads. An alternative function is supported. P0.1: PWM1 (14-Bit PWM Output)	3	
	P0.4-P0.7		General I/O Port (4-bit), Input or Output mode (push-pull or n-channel open drain) is software configurable.	7	
P1	P1.0-P1.3	I/O	Input/output mode or push-pull output mode is software configurable.	6	
	P1.4-P1.7		General I/O Port (4-bit), configurable for digital input or n-channel open drain output. P1.4-P1.7 can withstand up to 5-volt loads. Multiplexed for alternative use as external inputs ADC1-ADC4.	4	
P2	P2.0-P2.3	I/O	General I/O Port (4-bit), input or push-pull output mode is software configurable. Multiplexed for alternative use as external interrupt inputs INT0-INT3.	2	
	P2.4-P2.7		Input mode or push-pull output mode is software configurable. An alternative function is supported. P2.7: OSDHT (Halftone signal output)	6	
P3	P3.0	I/O	Input mode or push-pull output mode is software configurable.	6	

I/O PORT nDATA REGISTER (n = 0-3)

All four port data registers have the identical structure as shown in Figure 9-1 below:

Table 9-2. Port Data Register Summary

Register Name	Mnemonic	Address	Reset Value	R/W
Port 0 Data Register	P0	0x30030	xx00h	R/W
Port 1 Data Register	P1	0x30031	00xxh	R/W
Port 2 Data Register	P2	0x30032	xx00h	R/W
Port 3 Data Register	P3	0x30033	00xxh	R/W

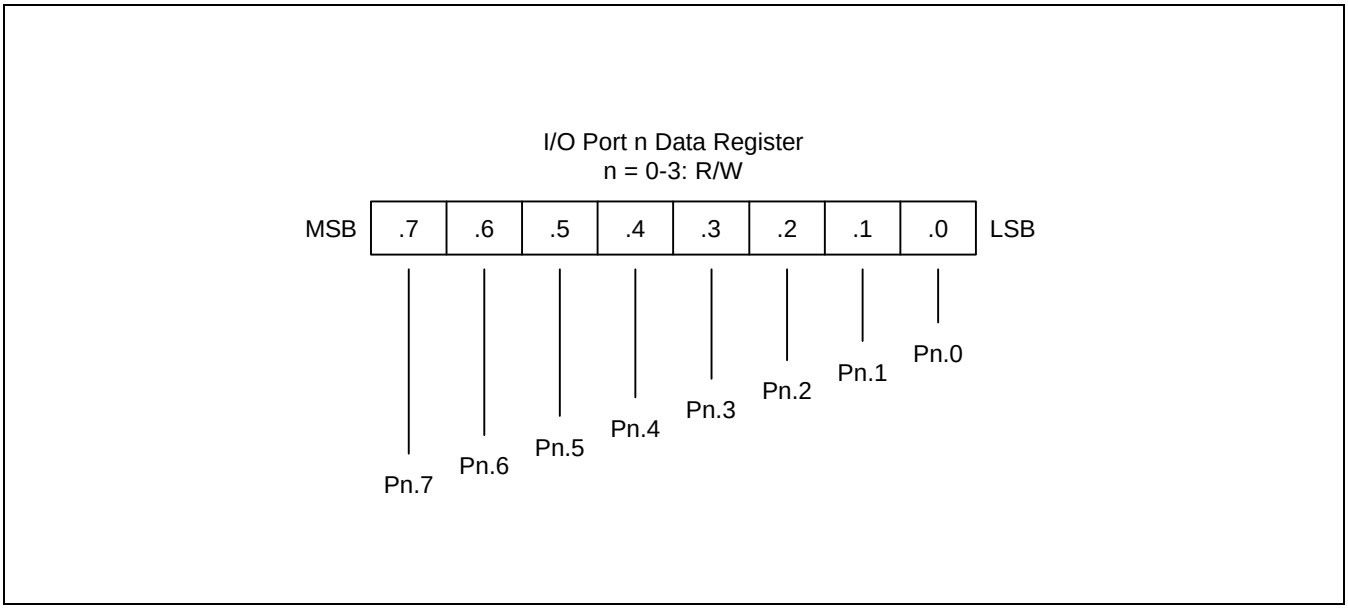


Figure 9-1. Port Data Register Structure

PORT 0 CONTROL REGISTER

Register	Offset Address	R/W	Description	Reset Value
P0CON	0x30034	R/W	Port 0 control register	0000h

Table 9-3. Port 0 Control Register

Name	Bit	Setting	Description
P0CON	1, 0	0 or 1	P0.0/PWM0 setting. 0x: Input mode 10: Output mode (push-pull type) 11: Alternative function mode (PWM0 function output)
	3, 2	0 or 1	P0.1/PWM1 setting. 0x: Input mode 10: Output mode (open-drain type) 11: Alternative function mode (PWM1 function output; open-drain type)
	5, 4	0 or 1	P0.2 setting. 0x: Input mode 10: Not used 11: Output mode (open-drain type)
	7, 6	0 or 1	P0.3 setting. 0x: Input mode 10: Not used 11: Output mode (open-drain type)
	9, 8	0 or 1	P0.4 setting. 0x: Input mode 10: Output mode (open-drain type) 11: Output mode (push-pull type)
	11, 10	0 or 1	P0.5 setting. 0x: Input mode 10: Output mode (open-drain type) 11: Output mode (push-pull type)
	13, 12	0 or 1	P0.6 setting. 0x: Input mode 10: Output mode (open-drain type) 11: Output mode (push-pull type)
	15, 14	0 or 1	P0.7 setting. 0x: Input mode 10: Output mode (open-drain type) 11: Output mode (push-pull type)

PORT 1 CONTROL REGISTER

Register	Offset Address	R/W	Description	Reset Value
P1CON	0x30036	R/W	Port 1 control register	0000h

Table 9-4. Port 1 Control Register

Name	Bit	Setting	Description
P1CON	1, 0	0 or 1	P1.0 setting 00: Input mode 01: Not used 10: Not used 11: Output mode (push-pull type)
	3, 2	0 or 1	P1.1 setting 00: Input mode 01: Not used 10: Not used 11: Output mode (push-pull type)
	5, 4	0 or 1	P1.2 setting 00: Input mode 01: Not used 10: Not used 11: Output mode (push-pull type)
	7, 6	0 or 1	P1.3 setting 00: Input mode 01: Not used 10: Not used 11: Output mode (push-pull type)
	9, 8	0 or 1	P1.4/ADC1 setting 00: Input mode 01: ADC input mode (digital input disable) 10: Not used 11: N-channel open-drain output mode (5 V load capacity)
	11, 10	0 or 1	P1.5/ADC2 setting 00: Input mode 01: ADC input mode (digital input disable) 10: Not used 11: N-channel open-drain output mode (5 V load capacity)
	13, 12	0 or 1	P1.6/ADC3 setting 00: Input mode 01: ADC input mode (digital input disable) 10: Not used 11: N-channel open-drain output mode (5 V load capacity)
	15, 14	0 or 1	P1.7/ADC4 setting 00: Input mode 01: ADC input mode (digital input disable) 10: Not used 11: N-channel open-drain output mode (5 V load capacity)

PORT 2 CONTROL REGISTER

Register	Offset Address	R/W	Description	Reset Value
P2CON	0x30038	R/W	Port 2 control register	0000h

Table 9-5. Port 2 Control Register

Name	Bit	Setting	Description
P2CON	1, 0	0 or 1	P2.0/INT0 setting 00: Input mode, interrupt disabled 01: Input mode, interrupt on rising edge 10: Input mode, interrupt on falling edge 11: Output mode (push-pull type)
	3, 2	0 or 1	P2.1/INT1 setting 00: Input mode, interrupt disabled 01: Input mode, interrupt on rising edge 10: Input mode, interrupt on falling edge 11: Output mode (push-pull type)
	5, 4	0 or 1	P2.2/INT2 setting 00: Input mode, interrupt disabled 01: Input mode, interrupt on rising edge 10: Input mode, interrupt on falling edge 11: Output mode (push-pull type)
	7, 6	0 or 1	P2.3/INT3 setting 00: Input mode, interrupt disabled 01: Input mode, interrupt on rising edge 10: Input mode, interrupt on falling edge 11: Output mode (push-pull type)
	9, 8	0 or 1	P2.4 setting 00: Input mode 01: Not used 10: Not used 11: Output mode (push-pull type)
	11, 10	0 or 1	P2.5 setting 00: Input mode 01: Not used 10: Not used 11: Output mode (push-pull type)
	13, 12	0 or 1	P2.6 setting 00: Input mode 01: Not used 10: Not used 11: Output mode (push-pull type)
	15, 14	0 or 1	P2.7/OSDHT setting 00: Input mode 01: Not used 10: OSDHT output mode 11: Output mode (push-pull type)

PORT 3 CONTROL REGISTER

Register	Offset Address	R/W	Description	Reset Value
P3CON	0x3003A	R/W	Port 3 control register	0000h

Table 9-6. Port 3 Control Register

Name	Bit	Setting	Description
P3CON	1, 0	0 or 1	P3.0 setting 00: Input mode 01: Not used 10: Not used 11: Output mode (push-pull type)
	15, 2	–	Not used

10 REAL TIMER

OVERVIEW

Real time clock control register represents the information of one second timing, proving an interrupt in every second.

If RTCON.0 = “1”, the real time clock counter is cleared, and the counting starts from 0 again. Under the state where the real time clock is enabled (TRCON.1 = “1”), when RTCON.2 bit is set to “1”, a real time interrupt is generated in every second. Occurring in sleep mode, a real time interrupt can release the mode.

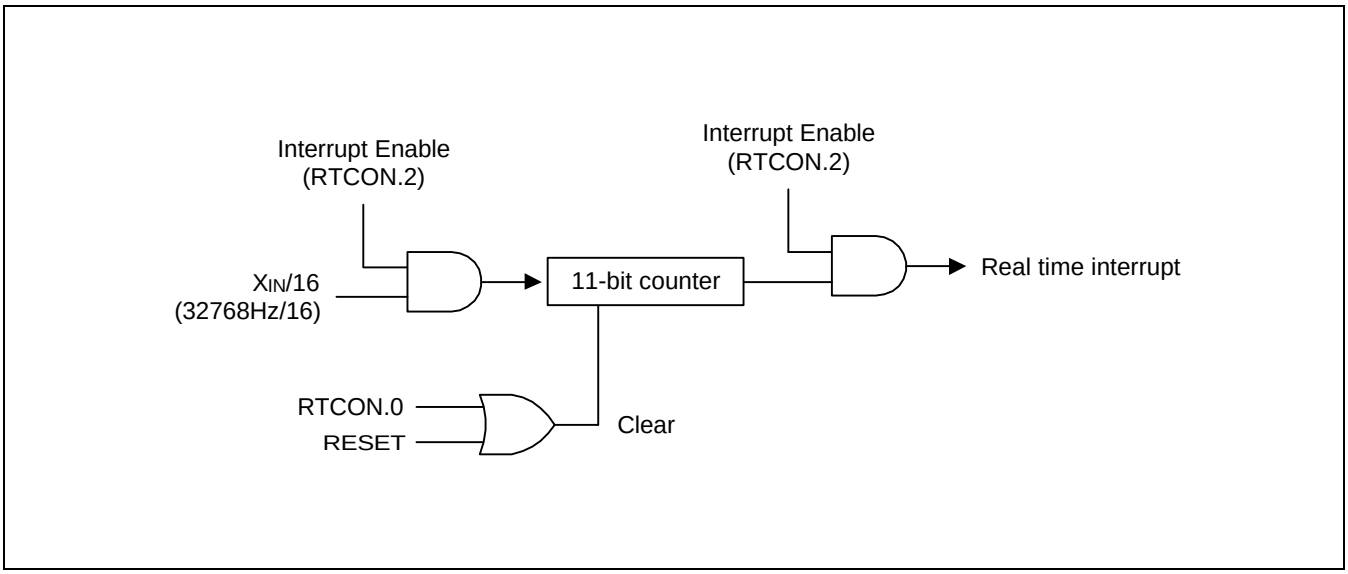


Figure 10-1. Real Timer Block Diagram

REAL TIME CLOCK CONTROL REGISTER

Register	Offset Address	R/W	Description	Reset Value
RTCON	0x3001C	R/W	Real timer clock control register	0000h

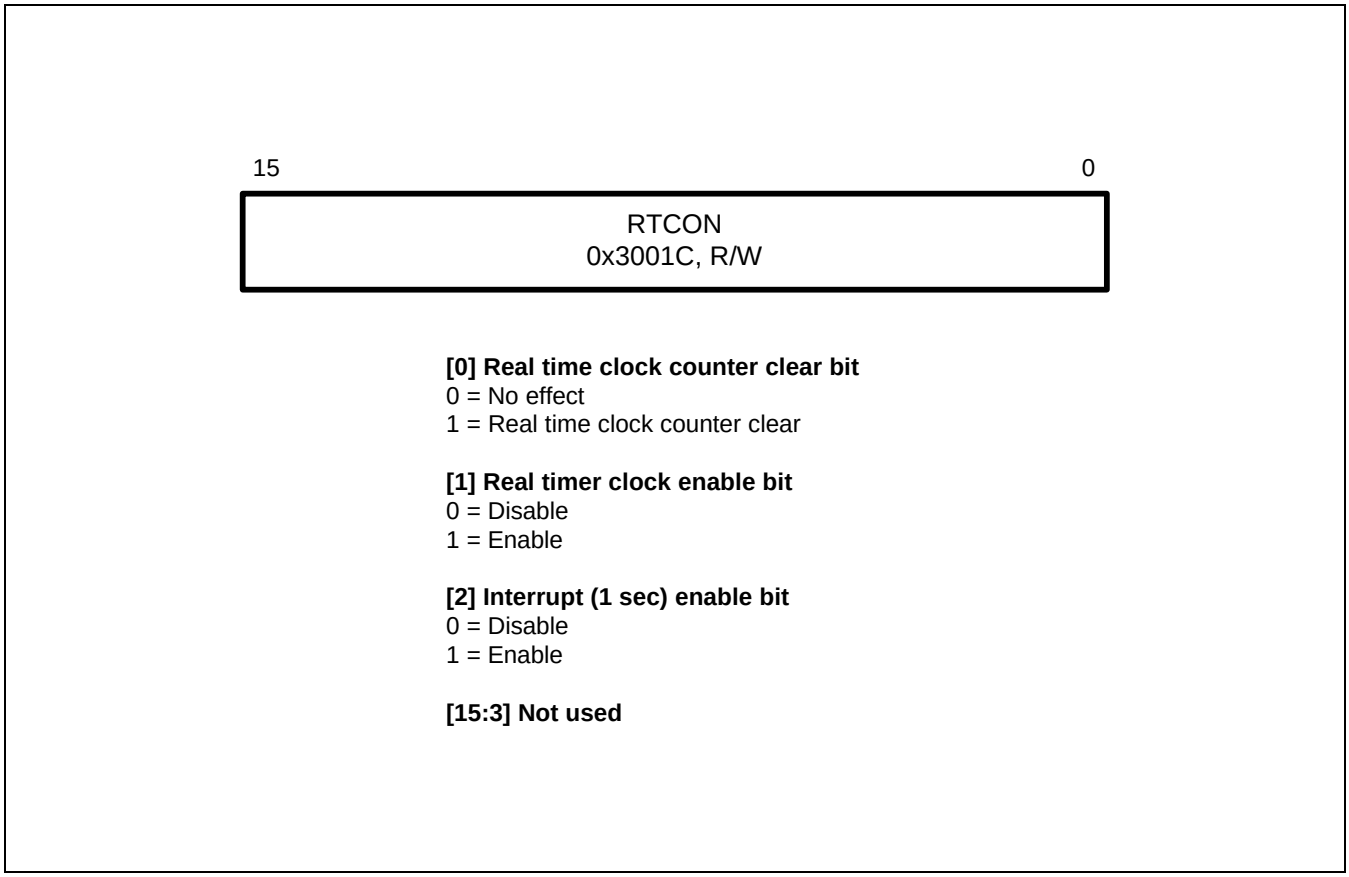
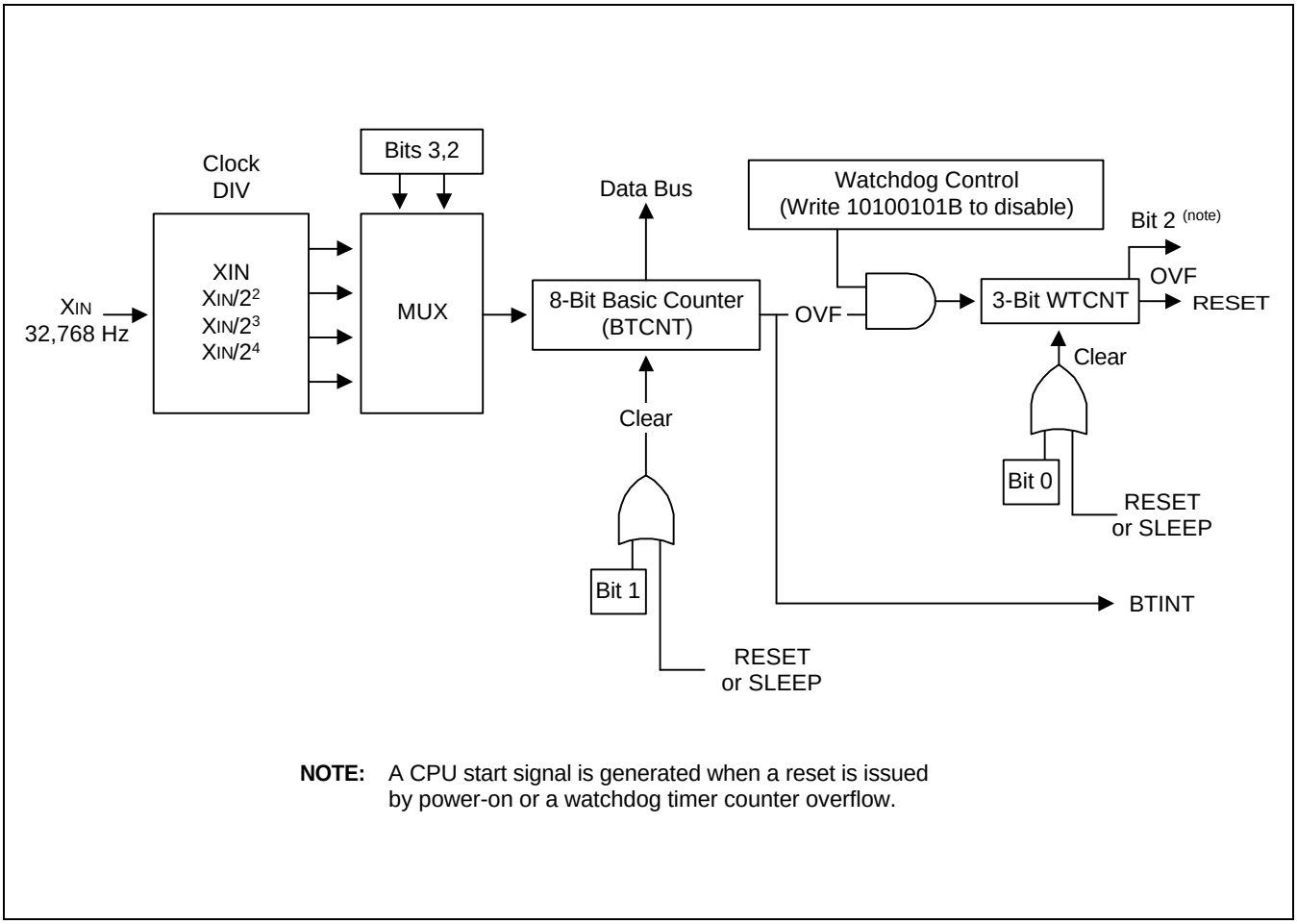


Figure 10-2. Real Time Clock Control Register (RTCON)

11 BASIC TIMER & WATCHDOG TIMER

OVERVIEW

The S3C380D/F380D Basic Timer/Watchdog Timer is used to resume the controller operation when it is disturbed by noise or other kinds of system error or malfunctions. It can be used as a normal interval timer to request interrupt services. It also signals the end of the required oscillation interval after a reset or a Sleep mode release. Users can control the disable or enable value for watchdog timer in BTCON.



NOTE: A CPU start signal is generated when a reset is issued by power-on or a watchdog timer counter overflow.

Figure 11-1. Basic and Watchdog Timer Block Diagram

BASIC TIMER COUNTER REGISTER

The basic timer counter register, BTCNT (address: 0x30002), is used to specify the time out duration, and is a free run 8-bit counter.

Register	Offset Address	R/W	Description	Reset Value
BTCNT	0x30002	R	Basic Timer Count register	xx00h

Table 11-1. Basic Timer Counter Setting (at Xin = 32,768 Hz)

BTCON.3	BTCON.2	Clock source	Resolution	Interval Time	Max. interval	Remark
0	0	$X_{IN}/2^4$	488.288 uS	$2^4/X_{IN}$ x 2^8	125.0 mS	Default setting
0	1	$X_{IN}/2^3$	244.144 uS	$2^3/X_{IN}$ x 2^8	62.50 mS	–
1	0	$X_{IN}/2^2$	122.072 uS	$2^2/X_{IN}$ x 2^8	31.25 mS	–
1	1	X_{IN}	30.518 uS	$1/X_{IN}$ x 2^8	7.812 mS	–

WATCHDOG TIMER COUNTER REGISTER

The watchdog timer counter register, WTCNT, is a free run 3-bit counter, used to specify the time out duration. The watchdog timer enable data can be any value, except 0xA5xx, to make the watchdog timer overflow, which will issue a system hardware reset.

Watchdog timer counter register, WTCNT, can not be read.

Table 11-2. Watchdog Timer Counter Setting (at 32,768 Hz)

BTCON.3	BTCON.2	Clock source	Resolution	WDT interval.	Interval time	Remark
0	0	$X_{IN}/2^4$	488.288 uS	$2^4/X_{IN}$ x 2^8 x 2^3	1 S	Default setting
0	1	$X_{IN}/2^3$	244.144 uS	$2^3/X_{IN}$ x 2^8 x 2^3	0.5 S	–
1	0	$X_{IN}/2^2$	122.072 uS	$2^2/X_{IN}$ x 2^8 x 2^3	250 mS	–
1	1	X_{IN}	30.518 uS	$1/X_{IN}$ x 2^8 x 2^3	62.5 mS	–

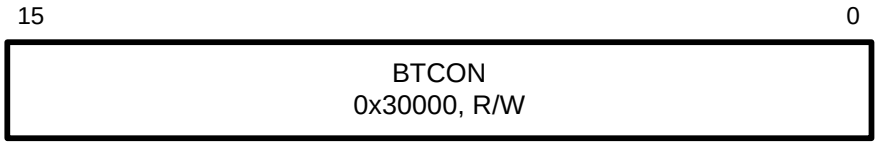
BASIC TIMER CONTROL REGISTER

The basic timer control register, BTCON, contains watchdog counter enable bits, clock input setting bits, and counter clear bits.

Registers	Address	R/W	Description	Reset Value
BTCON	0x30000	R/W	Basic and Watchdog Timer Control register	0000h

Basic timer control register has the following bits:

[0]	Watchdog Counter (WTCNT) clear bit	This bit clears the watch dog counter. The counter is loaded with all zero values.
[1]	Basic Counter (BTCNT) clear bit	This bit clears the basic counter. When this bit is set, it automatically will be cleared after the counter is loaded with all zero values.
[3:2]	Clock source select	These bits select a clock source. When it is 11B, it selects an XIN as a clock source. When 10B, an $X_{IN}/2^2$, when 01B, an $X_{IN}/2^3$, and, when 00B, an $X_{IN}/2^4$.
[7:4]	No effect	
[15:8]	Watch dog timer enable	These bits control enabling or disabling a watchdog timer counting. When these bits are {10100101} value, the watchdog timer counter stops. Other values can enable the counting of the watchdog timer, and make the system reset when an overflow signal occurs. Reset value is 0x0000. After a reset, the watchdog counter is enabled.



[0] Watchdog Counter (WTCNT) Clear Bit
0 = No effect
1 = Watchdog counter clear

[1] Basic Counter (BTCNT) Clear Bit
0 = No effect
1 = Basic counter clear

[3:2] Clock Source Select
00 = $X_{IN}/2^4$
01 = $X_{IN}/2^3$
10 = $X_{IN}/2^2$
11 = X_{IN}

[7:4] No Effect

[15:8] Watchdog Timer Enable
10100101 = Disable
Others = Enable

Figure 11-2. Basic Timer Control Register

FUNCTION DESCRIPTION

INTERVAL TIMER FUNCTION

The basic timer's primary function is to measure elapsed time intervals. The standard time interval is equal to 256 basic timer clock pulses.

The 8-bit counter register (BTCNT) is incremented each time a clock signal is detected that corresponds to the frequency selected by BTCON. BTCNT continues incrementing as it counts BT clocks until an overflow occurs (255). An overflow causes the Basic timer interrupt pending flag to be set to logic one to signal that the designated time interval has elapsed. An interrupt request is then generated, BTCNT is cleared to logic zero, and counting continues from 00H.

Oscillation Stabilization Interval Timer Function

You can also use the basic timer to program a specific oscillation stabilization interval following a RESET or when SLEEP mode has been released by external interrupt0, 1, 2, 3, V-sync interrupt, remocon signal input interrupt or real timer interrupt.

In SLEEP mode, whenever a RESET or SLEEP release interrupt occurs, the PLL block operation starts.

Oscillator DOES NOT stop in SLEEP mode.

The BTCNT value then starts increasing at the rate of $X_{IN}/2^4$ (for reset), or at the rate of the $X_{IN}/1$ (for SLEEP released by interrupt). When bit 2 (WTCNT.2) of 3-bit Watchdog timer counter (WTCNT) is set (for reset), or bit7 (BTCNT.7) of 8bit Basic timer counter (BTCNT) is set (for SLEEP released by interrupt), a signal is generated to indicate that the stabilization interval has elapsed and to gate the clock signal off to the CPU so that it can resume the normal operation.

In summary, the following events occur when SLEEP mode is released by INTERRUPT:

1. During SLEEP mode, external interrupt0, 1, 2, 3, V-sync interrupt, remocon signal input interrupt or real timer interrupt occurs to trigger a SLEEP mode release and PLL block starts operation. But Oscillator DOES NOT stop in SLEEP mode.
2. When SLEEP mode is released, the BTCNT value increases at the rate of the $X_{IN}/1$.
3. Clock oscillation stabilization interval begins and continues until bit7 (BTCNT.7) of 8-bit Basic timer is set (that is 3.90625 ms).
4. When bit7 (BTCNT.7) of 8-bit Basic timer is set, the normal CPU operation resumes.

In summary, the following events occur when SLEEP mode is released by RESET:

1. During SLEEP mode, a power-on reset releases the SLEEP mode and PLL block starts. But Oscillator DOES NOT stop in SLEEP mode.
2. If a reset occurs in SLEEP mode, the basic timer counter will increase at the rate of $X_{IN}/2^4$.
3. Clock oscillation stabilization interval begins and continues until bit2 (WTCNT.2) of 3-bit Watchdog timer (WTCNT) is set (that is 500 ms). When bit 2 of 3-bit watchdog timer is set, the normal CPU operation resumes.

In summary, the following events occur when oscillation is stabilized by POWER-ON RESET:

1. A power-on reset activates the oscillator which makes PLL block start.
2. If a power-on reset occurred, the basic timer counter will increase at the rate of $X_{IN}/2^4$.
3. Clock oscillation stabilization interval begins and continues until bit 2 (WTCNT.2) of 3-bit Watchdog timer (WTCNT) is set (that is 500 ms).
4. When bit 2 of 3-bit Watchdog timer is set, the normal CPU operation resumes.

Watchdog Timer Function

The basic timer can also be used as a "watchdog" timer to detect inadvertent program loop, that is, a system or program operation error. For this purpose, an instruction that clears the watchdog timer within a given period should be executed at proper points in the program. If an instruction that clears the watchdog timer is not performed within the period and the watchdog timer overflows, RESET signal is generated and the system is reset.

The operation of the watchdog timer is as follows:

1. Each time BTCNT overflows, an overflow signal is sent to the watchdog timer counter, WTCNT.
2. If 3-bit WTCNT overflows, a system reset is generated.

12 16-BIT TIMERS

OVERVIEW

The S3C380D/F380D has three 16-bit timers: T0, T1, and T2. These timers can operate in interval mode. The clock source for the timers can be an internal clock. You can enable or disable the timers by setting control bits in the corresponding timer mode register.

TIMER CONTROL REGISTER (TMCON)

You can use the timer control register, TMCON, to

— Select the timer n enable (stop or run)

— Select the timer n input clock selection

— Timer n interrupt enable or disable

IntMask register controls enabling or disabling the timer n overflow interrupt. IntPend register contains timer n match interrupt pending bits.

A reset clears TMCON to '0000H'. This sets the timer n disable, selects an input clock frequency of $f_{CPU}/1000$ and disables all timer n interrupts.

INTERVAL MODE OPERATION

In interval timer mode, an overflow signal is generated when the counter value written to the Tn reference data register, TnDATA, is decreased, generating on overflow. The overflow signal generates a timer n interrupt (TnINT) and reloads TnDATA to TnCNT.

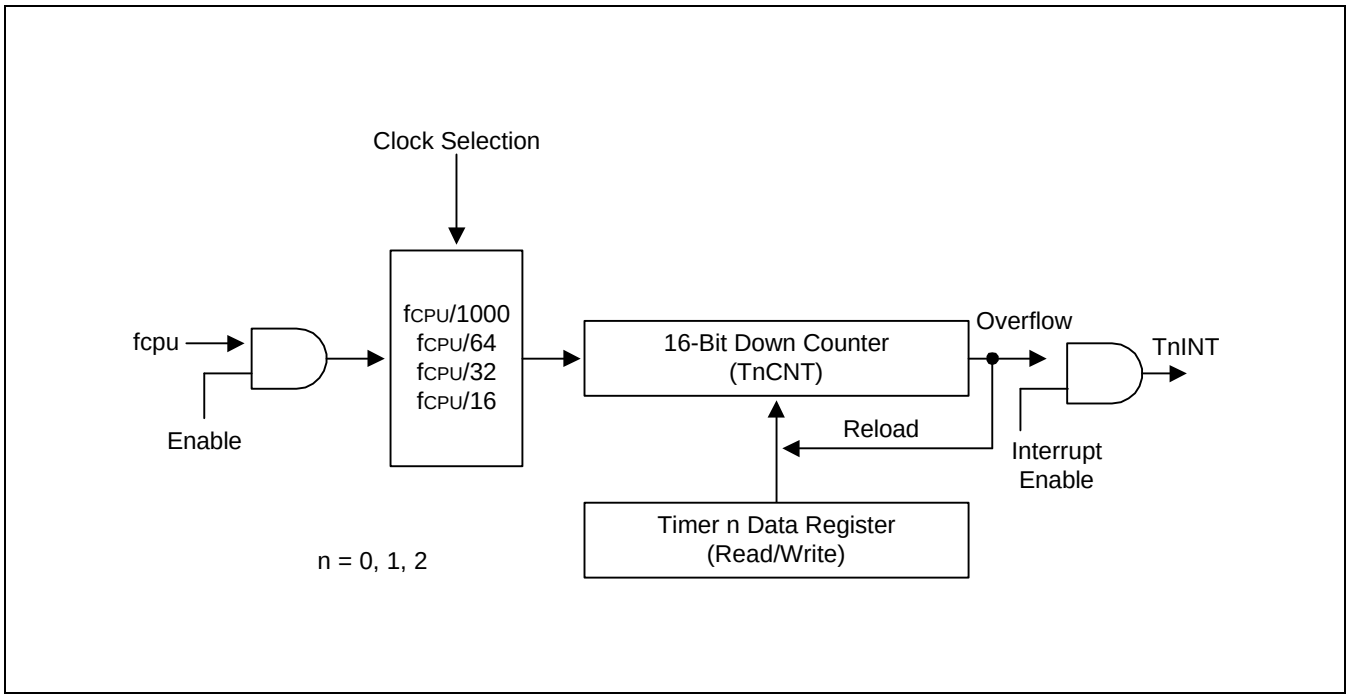


Figure 12-1. 16-Bit Timer Block Diagram

TIMER SPECIAL REGISTERS

TIMER CONTROL REGISTERS

The timer control register, TMCON, is used to control the operation of the three 16-bit timers.

Table 12-1. Timer Control Register Description

Register	Offset Address	R/W	Description	Reset Value
TMCON	0x3000A	R/W	Timer control register	0000h

Three timer mode registers have the following control settings:

[0]	Timer 0 enable	This bit enables or disables timer 0. When the bit is set to “0”, the 16-bit is set to “0”, the 16-bit timer 0 down counter stops. When it is “1”, the 16-bit timer 0 down counter starts counting again and the counter value decrements by one on accepting every clock.
[2:1]	Timer 0 clock selection	This field represents the clock source entering the 16-bit timer 0 down counter. The values, 00B, 01B, 10B, and 11B represent the clock source of $f_{CPU}/1000$, $f_{CPU}/64$, $f_{CPU}/32$, and $f_{CPU}/16$, respectively. Following a reset, $f_{CPU}/1000$ is selected.
[3]	Timer 0 interrupt enable bit	This bit controls timer 0 interrupt enable/disable. The value of the 16-bit timer 0 down counter decrements by one on accepting every clock. When an overflow occurs, the value of the timer 0 data register is reloaded to T0CNT, and if timer 0 interrupt is set enabled, an interrupt is generated.
[4]	Timer 1 enable	This bit enables or disables timer 1. When the bit is set to “0”, the 16-bit is set to “0”, the 16-bit timer 1 down counter stops. When it is “1”, the 16-bit timer 1 down counter starts counting again and the counter value decrements by one on accepting every clock.
[6:5]	Timer 1 clock selection	This field represents the clock source entering the 16-bit timer 1 down counter. The values, 00B, 01B, 10B, and 11B represent the clock source of $f_{CPU}/1000$, $f_{CPU}/64$, $f_{CPU}/32$, and $f_{CPU}/16$, respectively. Following a reset, $f_{CPU}/1000$ is selected.
[7]	Timer 1 interrupt enable bit	This bit controls timer 1 interrupt enable/disable. The value of the 16-bit timer 1 down counter decrements by one on accepting every clock. When an overflow occurs, the value of the timer 1 data register is reloaded to T1CNT, and if timer 1 interrupt is set enabled, an interrupt is generated.

[8]	Timer 2 enable	This bit enables or disables timer 2. When the bit is set to “0”, the 16-bit is set to “0”, the 16-bit timer 2 down counter stops. When it is “1”, the 16-bit timer 2 down counter starts counting again and the counter value decrements by one on accepting every clock.
[10:9]	Timer 2 clock selection	This field represents the clock source entering the 16-bit timer 2 down counter. The values, 00B, 01B, 10B, and 11B represent the clock source of $f_{CPU}/1000$, $f_{CPU}/64$, $f_{CPU}/32$, and $f_{CPU}/16$, respectively. Following a reset, $f_{CPU}/1000$ is selected.
[11]	Timer 2 interrupt enable bit	This bit controls timer 2 interrupt enable/disable. The value of the 16-bit timer 2 down counter decrements by one on accepting every clock. When an overflow occurs, the value of the timer 2 data register is reloaded to T2CNT, and if timer 2 interrupt is set enabled, an interrupt is generated.
[15:12]	Not used	

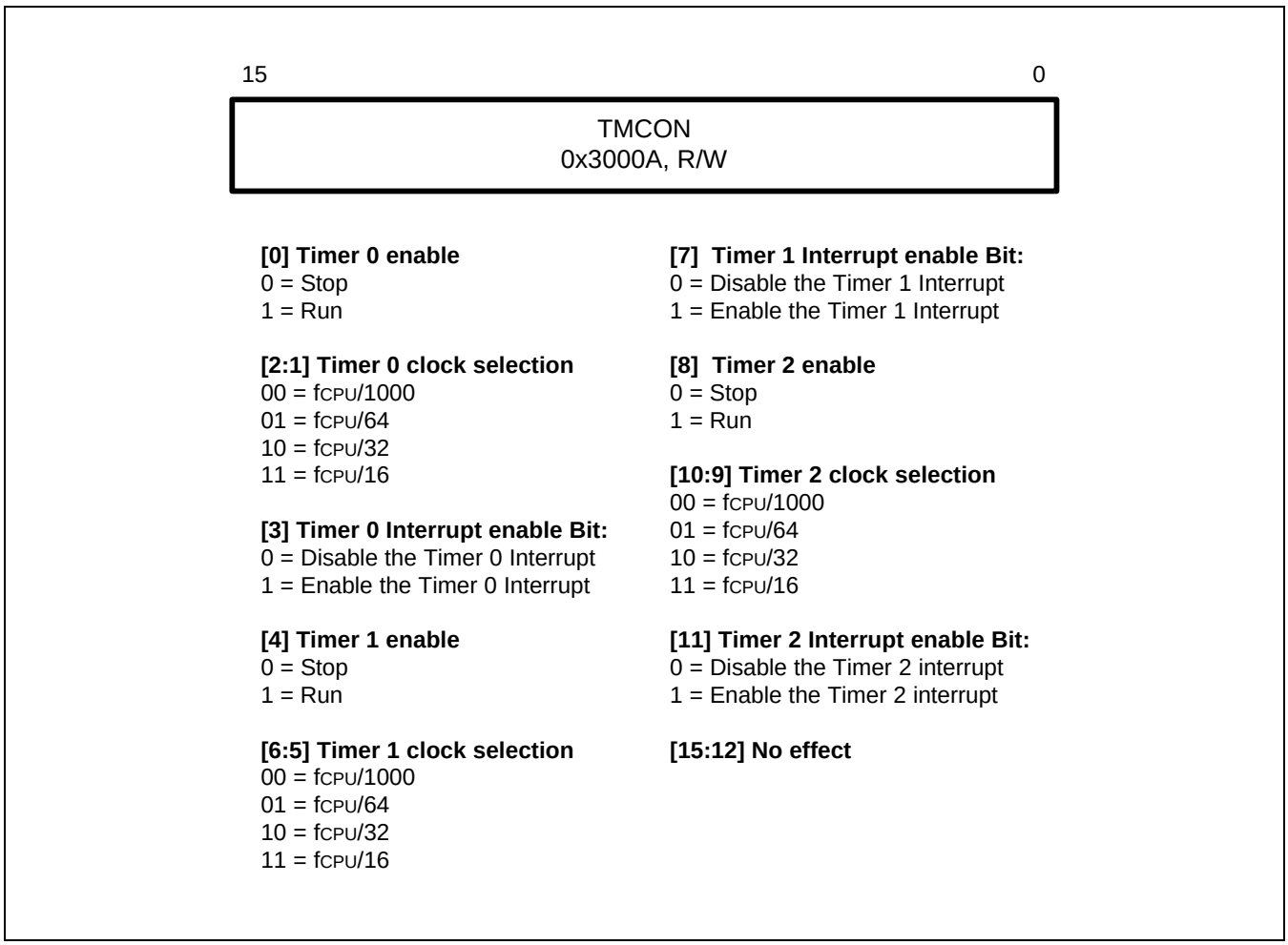


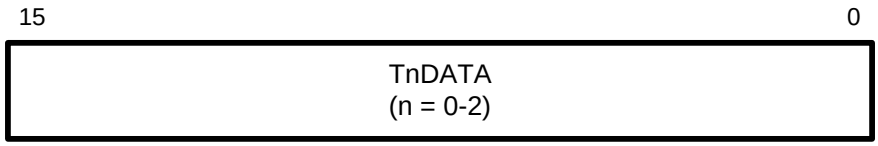
Figure 12-2. Timer Control Register

TIMER DATA REGISTERS

The timer data registers, T0DATA, T1DATA, and T2DATA, contain a value that specifies the time-out duration for each timer. The formula for calculating time-out duration is (Timer data + 1) cycles. See Figure 12-3 below.

Table 12-2. Timer Data Registers Description

Register	Offset Address	R/W	Description	Reset Value
T0DATA	0x30004	R/W	Timer 0 data register	ffffh
T1DATA	0x30006	R/W	Timer 1 data register	ffffh
T2DATA	0x30008	R/W	Timer 2 data register	ffffh



[15:0] Timer data value
This field specifies the time-out period of the corresponding timer. The time-out period is calculated as (Timer data + 1) cycles. Therefore, a maximum time-out period of 2^{16} cycles is possible (when the timer data value is 0xffff). The minimum time-out period (2 cycles) is obtained by writing the value 0x0001h to the timer data register field.

NOTE: Because the timer data register reset value is FFFFH, time is counted for the selected fcpu clock x FFFFH when the initial timer counter is enabled after reset. Because the timer data value is loaded to the timer counter when the timer counter overflows from 0 -> FFFFH, the initial timer data value, after reset, is updated when the timer data register's reset value, FFFFH, overflows from 0 to FFFFH. Therefore, after the timer counter is enabled, the overflow time, different from the one set by the user, is set until the initial overflow.

Figure 12-3. Timer Data Registers (T0DATA, T1DATA, and T2DATA)

TIMER COUNT REGISTERS

The timer count registers, T0CNT, T1CNT, and T2CNT, contain current timer 0, 1, and 2 count value, respectively.

The timer count registers operate as a decrement counter.

Table 12-3. Timer Count Registers description

Register	Offset Address	R/W	Description	Reset Value
T0CNT	0x3000C	R	Timer 0 count register	0
T1CNT	0x3000E	R	Timer 1 count register	0
T2CNT	0x30010	R	Timer 2 count register	0

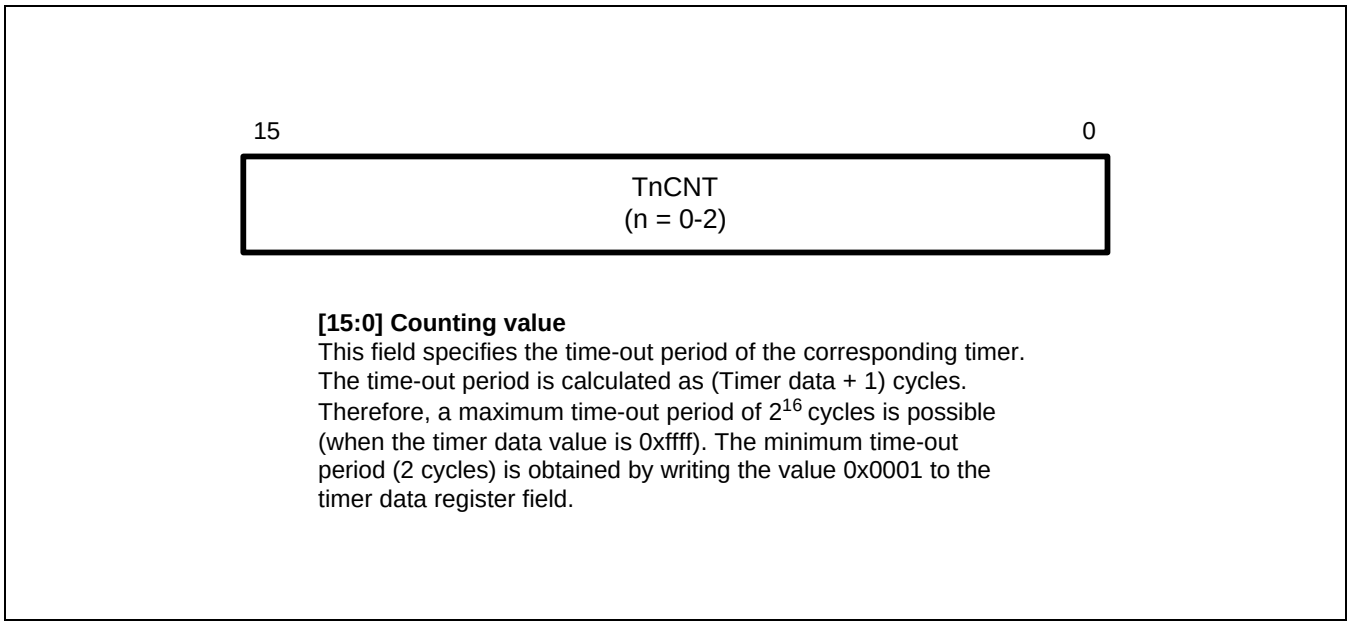


Figure 12-4. Timer Count Registers (T0CNT, T1CNT and, T2CNT)

13 PWM

OVERVIEW

The S3C380D microcontrollers have two 14-bit PWM circuits. The 14-bit circuits are called PWM0 and PWM1. The operation of all the PWM circuits is controlled by a single control register, PWMCON. PWMCON also contains a 4-bit prescaler for adjusting the PWM frequency (cycle).

The PWM counter is a 14-bit incrementing counter. It is used by the 14-bit PWM circuits. To start the counter and enable the PWM circuits, you must set PWMCON.4 to "1". If the counter is stopped, it retains its current count value; when re-started, it resumes counting from the retained count value.

A 4-bit prescaler controls the clock input frequency to the PWM counter. By modifying the prescaler value, you can divide the input clock by one to sixteen. The prescaler output is the clock frequency of the PWM counter.

PWM0–PWM1

The S3C380D pulse width modulation (PWM) module has two 14-bit PWM circuits (PWM0 and PWM1). The 14-bit PWM circuits have the following components:

— 14-bit counter with 4-bit prescaler (an 8-bit counter with 6-bit extension is used for 14-bit output resolution)

— 8-bit comparator and extension cycle circuit

— 8-bit reference data registers (Bit6-13 of PWM0,1)

— 6-bit extension data registers (Bit0-5 of PWM0,1)

— PWM output pins (PWM0, PWM1)

The PWM0 and PWM1 circuits are enabled by the PWMCON register.

PWM COUNTER

The PWM counter is a 14-bit increasing counter comprised of a 14-bit counter.

To determine the PWM module's base operating frequency, the upper byte counter is compared to the bit 6-13 of PWM data register value. In order to achieve higher resolutions, the lower six bits of the counter can be used to modulate the "stretch" cycle. To control the "stretching" of the PWM output duty cycle at specific intervals, the 6-bit high resolution counter value is compared with the 6-bit value (bits 0-5) that you write to the bit 0-5 of PWMn register.

PWM CLOCK RATE

The timing characteristics of both 14-bit output channels are identical, and are based on the maximum 16-MHz CPU clock frequency. The 4-bit prescaler value in the PWMCON register determines the frequency of the counter clock. You can set bit 0-3 of PWMCON to divide the CPU clock frequency by 1 (non-divided), 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, or 16.

Because the maximum CPU clock rate for the S3C380D microcontrollers is 16 MHz, the maximum base PWM frequency is 62.5 kHz (16 MHz divided by 256). This assumes a non-divided CPU clock.

PWM CONTROL REGISTER (PWMCON)

The control register for the PWM module, PWMCON, is located at the register address 0x30012. Bit settings in the PWMCON register control the following functions:

— 4-bit prescaler for scaling the PWM counter clock

— Stop/start (or resume) the PWM counter operation

A reset clears all PWMCON bits to logic zero, disabling the entire PWM module.

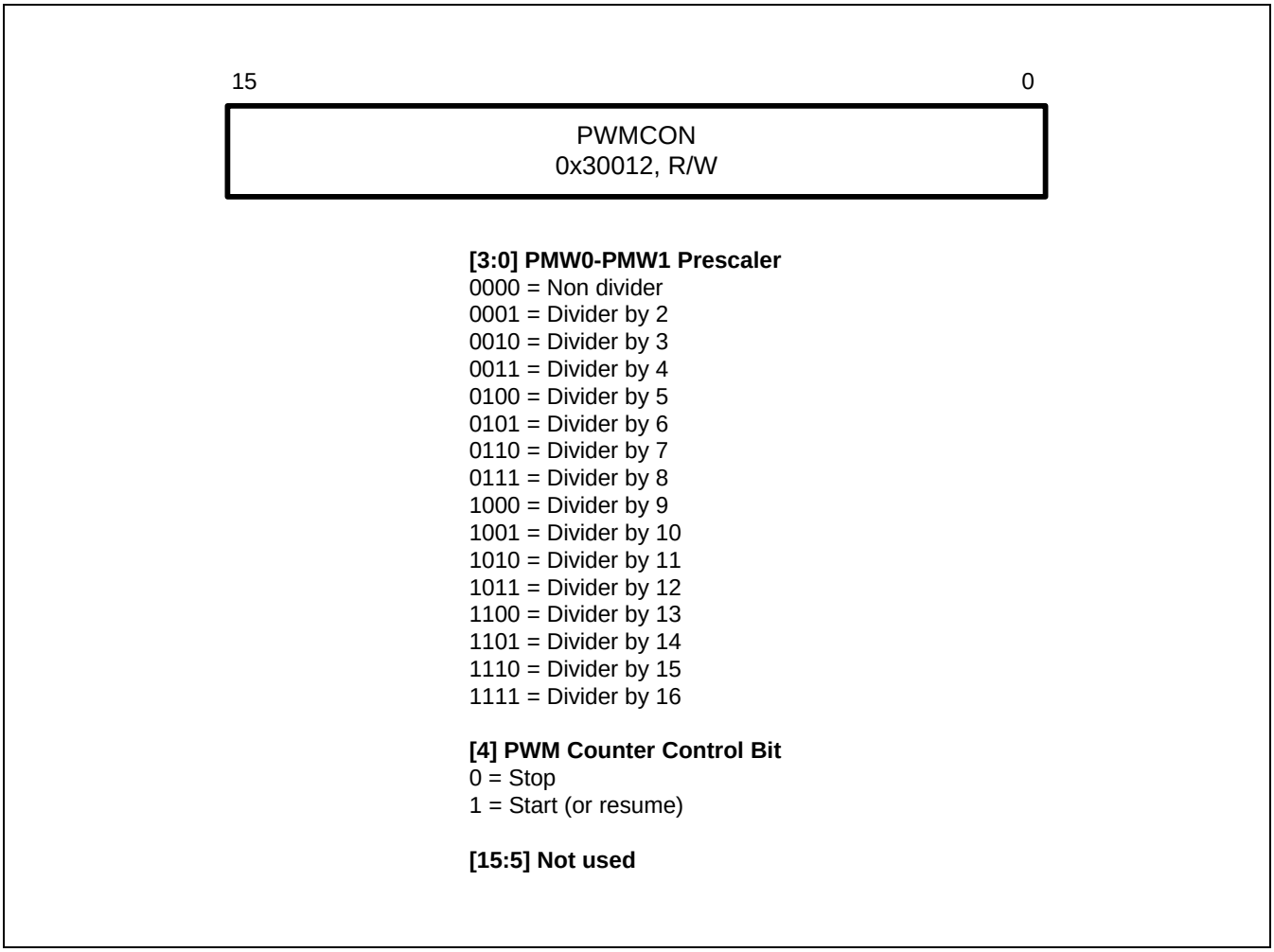


Figure 13-1. PWM Control Register (PWMCON)

PWM DATA REGISTERS (PWM0, PWM1)

Two PWM (duty) data registers determine the output value generated by each 14-bit PWM circuit. PWM0 and PWM1 are read/write addressable.

— 14-bit data registers PWM0 and PWM1

To program the required PWM output, you should load the appropriate initialization values into the 14-bit data registers (PWM0, PWM1). To start the PWM counter, or to resume counting, you should set PWMCON.4 to "1".

A reset operation disables all PWM output. The current counter value is retained when the counter stops. When the counter starts, counting resumes at the retained value.

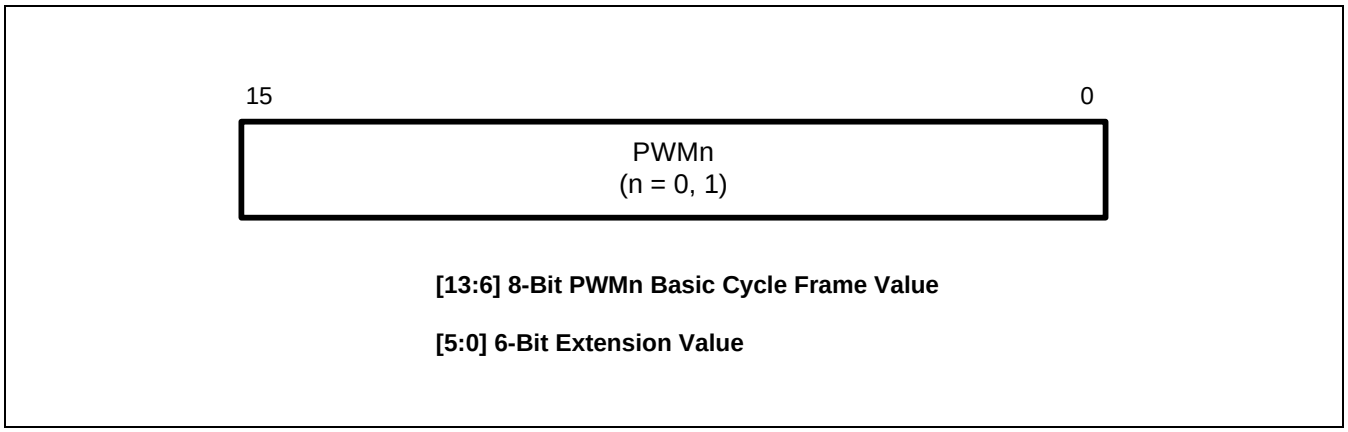


Figure 13-2. PWM Data Register (PWM0, PWM1)

Table 13-1. PWM0 and PWM1 Control and Data Registers

Register Name	Mnemonic	Address (Set 1, Bank 0)	Function
PWM0 data registers	Bit 13-6 of PWM0	0x30014	8-bit PWM0 basic cycle frame value
	Bit 5-0 of PWM0	0x30014	6-bit extension ("stretch") value
PWM1 data registers	Bit 13-6 of PWM1	0x30016	8-bit PWM1 basic cycle frame value
	Bit 5-0 of PWM1	0x30016	6-bit extension ("stretch") value
PWM control register	PWMCON	0x30012	PWM counter stop/start (resume), and 4-bit prescaler for CPU clock

PWM0 AND PWM1 FUNCTION DESCRIPTION

The PWM output signal toggles to Low level whenever the bit 13-6 of 14-bit counter matches the reference value stored in the bit 13-6 of PWM0 and PWM1, If the value in the bit 13-6 of PWM0 and PWM1 is not zero, an overflow of the bit 13-6 of 14-bit counter causes the PWM output to toggle to High level. In this way, the reference value written to the bit 13-6 of PWM0 and PWM1 determines the module's base duty cycle.

The value in the 6-bit extension counter (bit 5-0 of 14-bit counter) is compared with the extension settings in the 6-bit extension data register (bit 5-0 of PWM0 and PWM1). This 6-bit extension counter value, together with extension logic and the PWM module's extension register, is then used to "stretch" the duty cycle of the PWM output. The "stretch" value is one extra clock period at specific intervals, or cycles (see Table 13-2).

If, for example, the value in the extension register is '1', the 32nd cycle will be one pulse longer than the other 63 cycles. If the base duty cycle is 50%, the duty of the 32nd cycle will therefore be "stretched" to approximately 51% duty. For example, if you write 20H to the extension register, all odd-numbered pulses will be one cycle longer. If you write 3FH to the extension register, all pulses will be stretched by one cycle except the 64th pulse. PWM output goes to an output buffer and then to the corresponding PWM0 and PWM1 output pin. In this way, you can obtain high output resolution at high frequencies.

Table 13-2. PWM Output (Stretch) Values for Extension Registers PWM0 and PWM1

PWM0/1 Bit	(Stretched) Cycle Number
5	1, 3, 5, 7, 9, 11, ..., 55, 57, 59, 61, 63
4	2, 6, 10, 14, ... , 50, 54, 58, 62
3	4, 12, 20, ... , 44, 52, 60
2	8, 24, 40, 56
1	16, 48
0	32

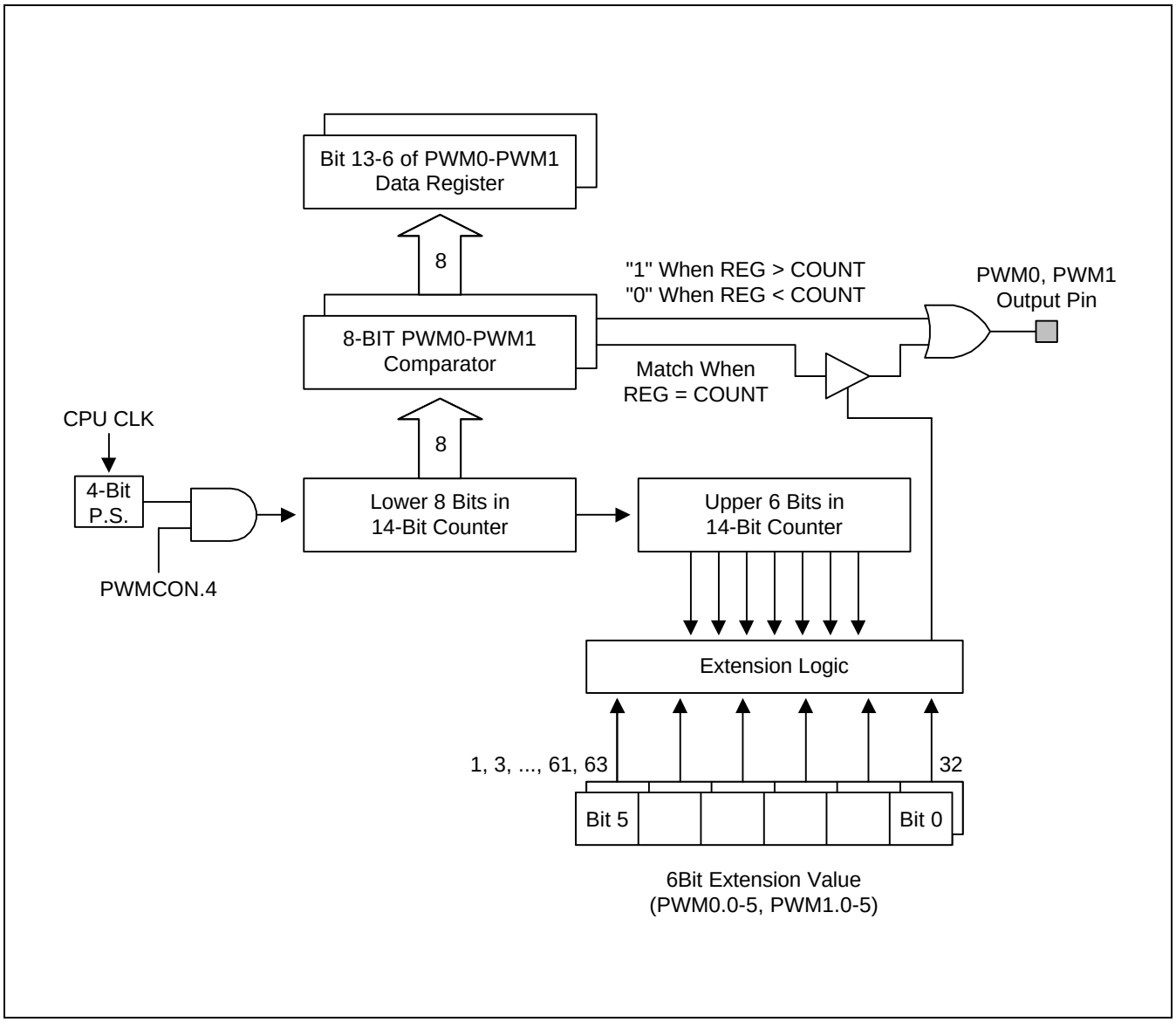


Figure 13-3. Block Diagram of 14-bit PWM Output Unit

14 REMOCON RECEIVER

OVERVIEW

The S3C380D/F380D has the ability to capture pulse signals which are externally input using H/W. It is able to transmit up to 8 different 8-bit capture data using the remocon receiver control register, RRCR, and FIFOD. The remocon receiver block operates with the remocon input interrupt bit, RRCR.10, set to “1”. It can choose rising edge, falling edge or rising/falling edge, and is able to choose the input pulse width from fvco/128, fvco/256, fvco/512, or 32,768 Hz for filtering. The frequency division of the 8-bit counter clock input is chosen from 1, 2, 4, or 8 of 32768 Hz. The FIFOs empty status flag is set when all FIFOs are empty, and the FIFOs full status flag is set when all FIFOs are full. An interrupt is generated when data is transmitted to all 8 FIFOs, and an 8-bit counter overflow interrupt occurs as well.

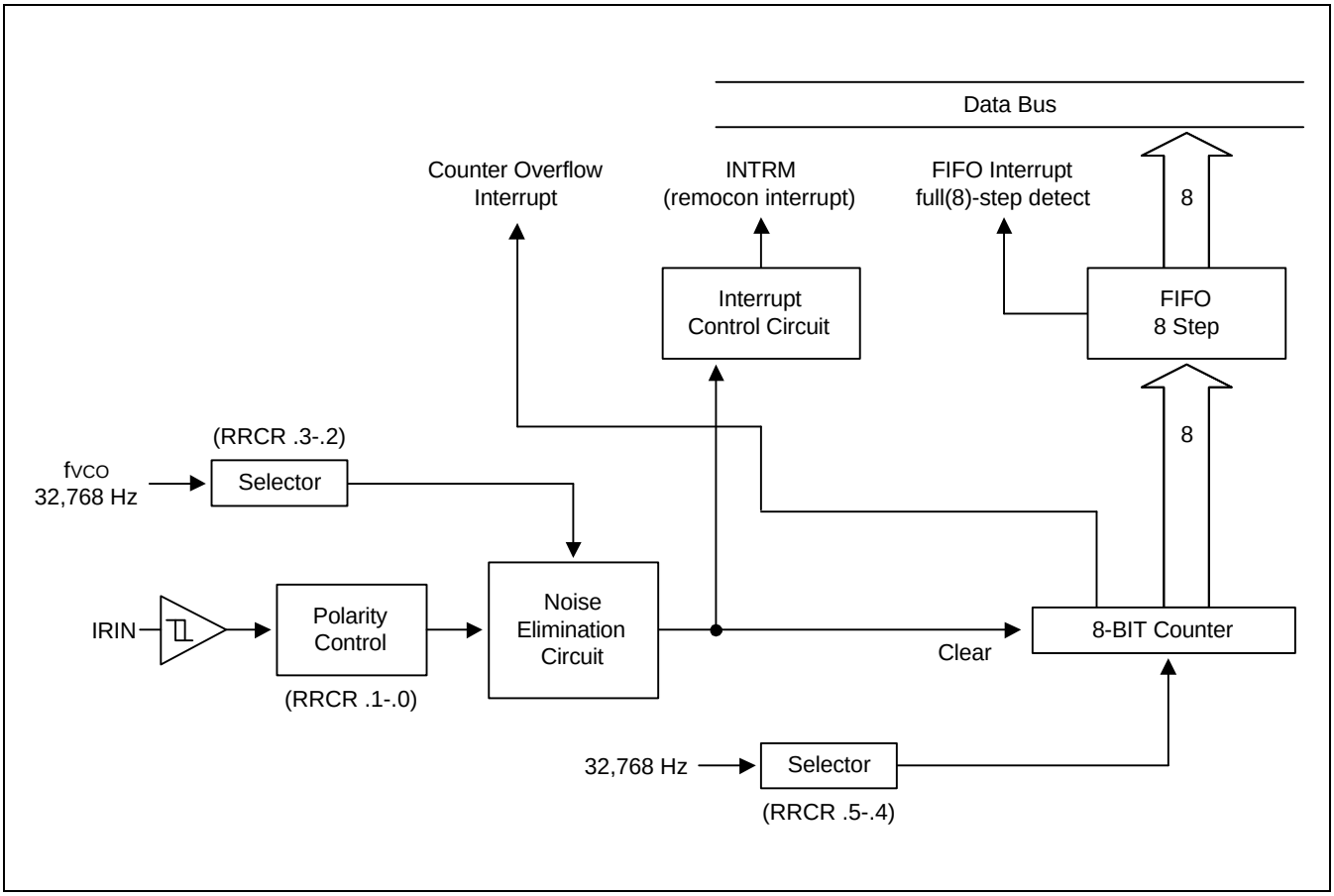


Figure 14-1. Remocon Receiver Circuit Block Diagram

Register	Offset Address	R/W	Description	Reset Value
RRCR	0x30018	R/W	Remocon receiver control register	0000h

[1,0]	Polarity control flag	00: Not used 01: Rising edge mode 10: Falling edge mode 11: Rising/Falling edge mode
[3,2]	Minimum pulse width recognized as remocon signal	Sampling clock Minimum remocon signal 00: fvco/128 01: fvco/256 10: fvco/512 11: 32,768 Hz
[5,4]	Counter clock selection	8-bit counter clock selection Maximum pulse width 00: 32,768 Hz/1 01: 32,768 Hz/2 10: 32,768 Hz/4 11: 32,768 Hz/8
[7,6]	No effect	
[8]	FIFO Empty Status flag	0: Not empty 1: Empty
[9]	FIFO full Status flag	0: Not full 1: Full
[10]	Remocon Input Interrupt	0: Disable the remocon input interrupt 1: Enable the remocon input interrupt
[11]	FIFO full (8)-step detect interrupt	0: Disable the FIFO full (8)-step detect interrupt 1: Enable the FIFO full (8)-step detect interrupt
[12]	Counter overflow interrupt	0: Disable the counter overflow interrupt 1: Enable the counter overflow interrupt
[15-13]	No effect	

NOTE: If all FIFOs are full, the next input data does not go into FIFO.

15 4-BIT ANALOG-TO-DIGITAL CONVERTER

OVERVIEW

The S3C380D/F380D has five 4-bit resolution A/D converter input pins(ADC0 to ADC4). The 4-bit A/D converter (ADC) module uses flash ADC logic to convert analog levels entering at one of the five input channels to equivalent 4-bit digital values. The analog input level must lie between the AV_{REF} and AV_{SS} values. The S3C380D/F380D has four normal ADC and one ADC (ADC0) for closed caption signal. The A/D converter has the following components:

— Analog comparator with thermometer decoder logic (flash ADC)

— D/A converter logic (resistor string type)

— ADC control register (ADCON)

— Five multiplexed analog data input pins (ADC0-ADC4)

— 4-bit A/D conversion data output (bit3-0 of ADCON)

— A/D Conversion time selection (ADCON.9-ADCON.8)

— A/D conversion star/stop control (ADCON.7)

FUNCTION DESCRIPTION

To initiate an analog-to-digital conversion procedure, you should write the channel selection data in the A/D converter control register, ADCON, and set the each port of ADC to ADC mode to select one of the five analog input pins (ADCn, n = 0-4) and set the conversion start bit, ADCON.7. The conversion loads data to bit 3-0 of ADCON register, and an INTAD interrupt can be generated.

The A/D conversion time can be chosen from 4, 8, 16, or 32 frequency division of the CPU clock, using the ADCON.9-8. The A/D conversion results are automatically dumped into ADCON.3-0. Once the ADC operation starts, it contains until the stopping signal is given by ADCON.7.

CONVERSION TIMING

The A/D conversion process requires 4 kinds of steps (f_{CPU}/4, f_{CPU}/8, f_{CPU}/16, and f_{CPU}/32). It uses the ADCON.9-8 to select the A/D conversion time from 4, 8, 16, or, 32 frequency division of the CPU clock.

In a 16 MHz CPU clock frequency, four clock cycle is 250 ns. If ADCON.9-.8 is '00B':

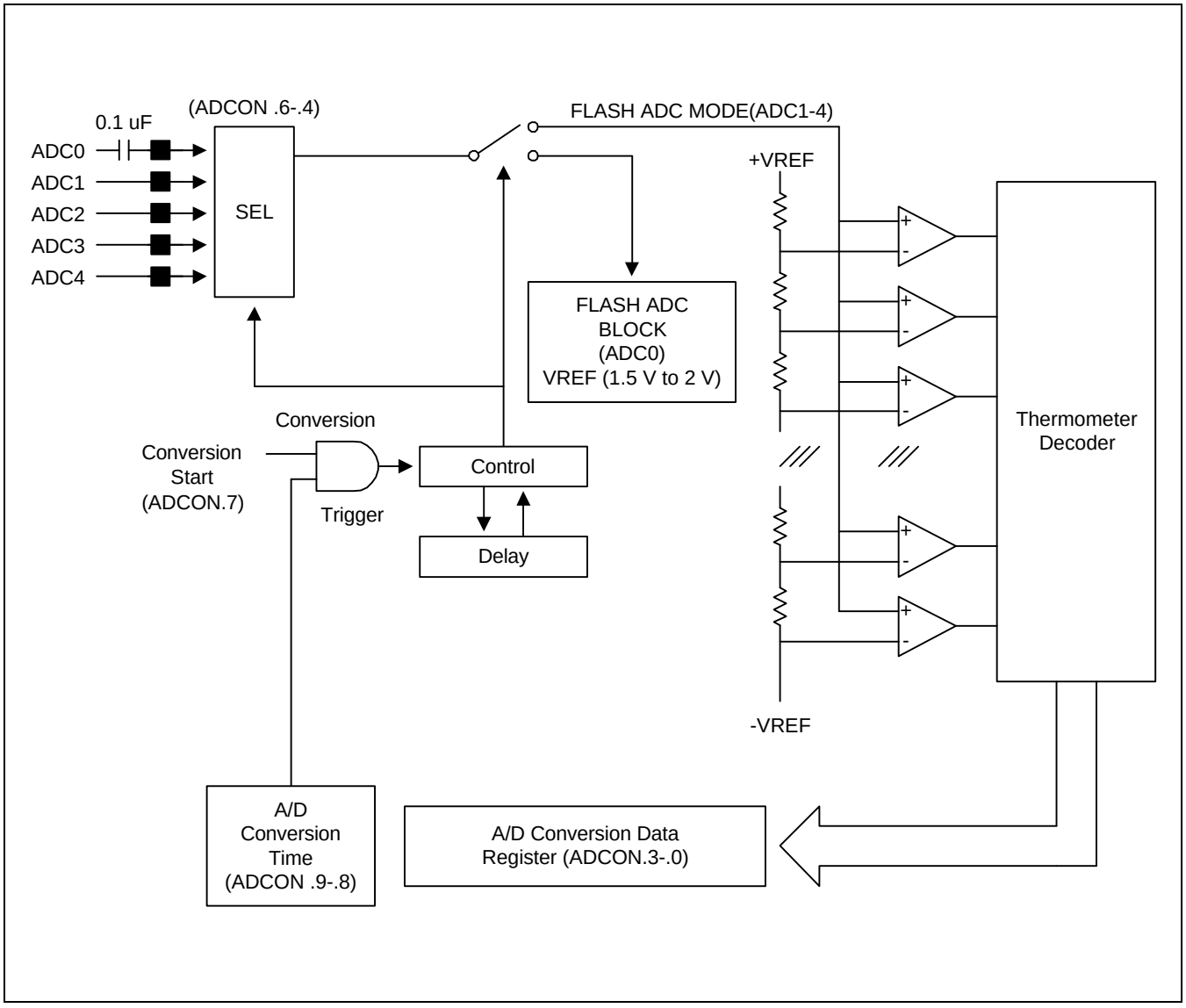


Figure 15-1. ADC Block Diagram

ADC SPECIAL REGISTERS

ADC CONTROL REGISTERS

The ADC control registers, ADCON, is used to control the operation of the five 4-bit A/DC channels.

Register	Offset Address	R/W	Description	Reset Value
ADCON	0x3003C	R/W (1)	ADC control register	x000h

[3:0] A/D Conversion Data (Read-only)

[4] Flash ADC Mode Selection
0 = Flash ADC0 mode
1 = Flash ADC1-4 mode

[6:5] A/D Conversion Selection
00 = Select ADC1
01 = Select ADC2
10 = Select ADC3
11 = Select ADC4

[7] A/D Conversion Control Bit
0 = Stop A/D conversion
1 = Starts A/D conversion

[9:8] A/D Conversion Time Selection
00 = $f_{CPU}/4$
01 = $f_{CPU}/8$
10 = $f_{CPU}/16$
11 = $f_{CPU}/32$

[15:10] Not used

NOTE: Bit3-bit0 of ADCON are read only.

Figure 15-2. ADC Control Register

Table 15-1. ADC Input Voltage Range

ADC Input	Input Voltage Range	Application
CVI/ADC0	**1.5V .. 2.0V**	**CCD Data Sampling Input**
P1.4/ADC1	0 .. V_{DD}	AFC Input/Key Scanning Input
P1.5/ADC2	0 .. V_{DD}	AFC Input/Key Scanning Input
P1.6/ADC3	0 .. V_{DD}	AFC Input/Key Scanning Input
P1.7/ADC4	0 .. V_{DD}	AFC Input/Key Scanning Input

16 ON-SCREEN DISPLAY (OSD)

OVERVIEW

The on-screen display (OSD/CCD) module displays channel number, the time, closed caption and other information on a display screen. The OSD character display module has 544 locations and supports a set of 256 characters. (One character is reserved: 00H for the blank function.) There are eight display colors.

PATTERN GENERATION SOFTWARE

For application development using the S3C380D/F380D microcontrollers, Samsung provides OSD pattern generation software (OSDFONT.exe). You can customize standard OSD patterns contained in this file.

Table 16-1. OSD Function Block Summary

OSD Function Block	Function Description
Video RAM	Located in 0x20000-0x2043F, the video RAM contains 544 "word" lines. Each line is 16 bits long. In OSD mode (OSDCON.0 = “0”), each 16-bit RAM stores an 8-bit character code, a character halftone or character background color mode selection bit, and italic character control bit. In CCD mode (OSDCON.0 = “1”), each 16-bit RAM store an 7-bit character code, a under time display, a blink display, a halftone display, a italic character display, a character background display control bit, and 3-bit character color or 2-bit character background color mode selection bit.
Character ROM	The character ROM contains an 18-dot × 16-dot matrix data for 256 characters. It is synchronized with the internal OSD clock. The ROM outputs the dot matrix data for each character. The function of one characters is pre-determined: 00H is used for blank (no-display) data.
Output control logic	Output control logic receives input from the Character ROM, OSD control registers, and fade control circuits. It then decides what to display on the screen and what color the display should be. On the basis of truth table calculations, the final OSD signals (blue, green, red, blank, and H/T) are output from the OSD block at pins 22–25, 21.

INTERNAL OSD CLOCK

Red-green-blue (RGB) color outputs, as well as display rates and positions, are determined by the clock signal, DOT_CLK. This signal is generated by the PLL block and is scaled by the dot and column counter. DOT_CLK equals the OSD oscillator clock divided by the clock divider value. The clock divider value is set by the horizontal character size settings in the OSDCON register.

The rate at which each new display line is generated is determined by H-sync input. The rate at which each new frame (screen) is generated is determined by V-sync input.

OSD VIDEO RAM

By setting OSDCON.0 to “0”, video RAM can be selected for OSD display. The OSD video RAM contains 544 word lines. Each line is 16 bits long. Of these 16 bits, eight are character display codes (bits 0–7).

Bit 8-10 allow you to choose the character color mode, and bit 11-12, allow you to choose the 4 character background color modes. Bit 13 is the Halftone control bit, and bit 14 is the character background enable bit. Bit 15 allows you to use italic characters.

CCD VIDEO RAM

By setting OSDCON.0 to “1”, video RAM can be selected for CCD display. The CCD video RAM contains 544 word lines. Each line is 16 bit long. Out of the 16-bits, bit 0-bit 6 choose the character code (128 words), and bit 7 controls the underline. Bit 8-10 choose from the 8 character color modes. Bit 11 enables the character background color, and bit 12 enables the character. Bit 13 enables the character's halftone, and bit 14 enables the character background. Bit 15 is the italic character enable bit.

DISPLAY POSITION CONTROL

The on-screen display has 544 character display positions. There are 34 horizontal columns and 16 vertical rows. Positions can be numbered sequentially from 0–543 (decimal) or from 0–21F (hexadecimal), as shown in Figures 16-1 and 16-2. To control display position, you can adjust the top and left margins and the inter-column and inter-row spacing between characters on the screen.

COLUMNS 0-33 →

DECIMAL

ROWS 0-15 ↓

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101
102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169
170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203
204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237
238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271
272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305
306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339
340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373
374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407
408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441
442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475
476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509
510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543

Figure 16-1. 544-Byte On-Screen Character Display Map (Decimal)

COLUMNS 0-33 →

HEXADECIMAL

ROWS 0-15 ↓

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21
22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40	41	42	43
44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65
66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80	81	82	83	84	85	86	87
88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB
CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED
EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF	100	101	102	103	104	105	106	107	108	109	10A	10B	10C	10D	10E	10F
110	111	112	113	114	115	116	117	118	119	11A	11B	11C	11D	11E	11F	120	121	122	123	124	125	126	127	128	129	12A	12B	12C	12D	12E	12F	130	131
132	133	134	135	136	137	138	139	13A	13B	13C	13D	13E	13F	140	141	142	143	144	145	146	147	148	149	14A	14B	14C	14D	14E	14F	150	151	152	153
154	155	156	157	158	159	15A	15B	15C	15D	15E	15F	160	161	162	163	164	165	166	167	168	169	16A	16B	16C	16D	16E	16F	170	171	172	173	174	175
176	177	178	179	17A	17B	17C	17D	17E	17F	180	181	182	183	184	185	186	187	188	189	18A	18B	18C	18D	18E	18F	190	191	192	193	194	195	196	197
198	199	19A	19B	19C	19D	19E	19F	1A0	1A1	1A2	1A3	1A4	1A5	1A6	1A7	1A8	1A9	1AA	1AB	1AC	1AD	1AE	1AF	1B0	1B1	1B2	1B3	1B4	1B5	1B6	1B7	1B8	1B9
1BA	1BB	1BC	1BD	1BE	1BF	1C0	1C1	1C2	1C3	1C4	1C5	1C6	1C7	1C8	1C9	1CA	1CB	1CC	1CD	1CE	1CF	1D0	1D1	1D2	1D3	1D4	1D5	1D6	1D7	1D8	1D9	1DA	1DB
1DC	1DD	1DE	1DF	1E0	1E1	1E2	1E3	1E4	1E5	1E6	1E7	1E8	1E9	1EA	1EB	1EC	1ED	1EE	1EF	1F0	1F1	1F2	1F3	1F4	1F5	1F6	1F7	1F8	1F9	1FA	1FB	1FC	1FD
1FE	1FF	200	201	202	203	204	205	206	207	208	209	20A	20B	20C	20D	20E	20F	210	211	212	213	214	215	216	217	218	219	21A	21B	21C	21D	21E	21F

Figure 16-2. 544-Byte On-Screen Character Display Map (Hexadecimal)

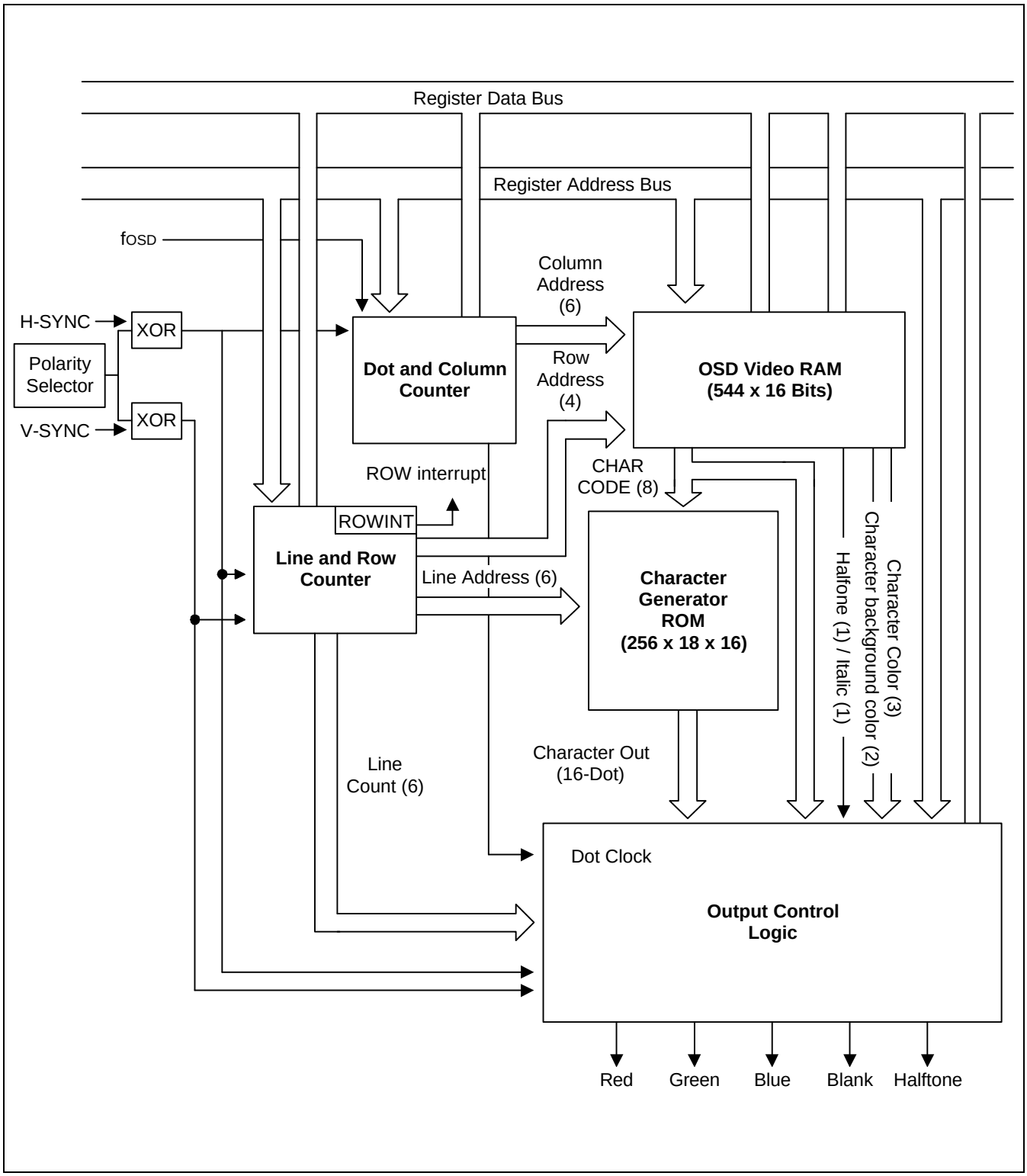


Figure 16-3. On-Screen Display Function Block Diagram

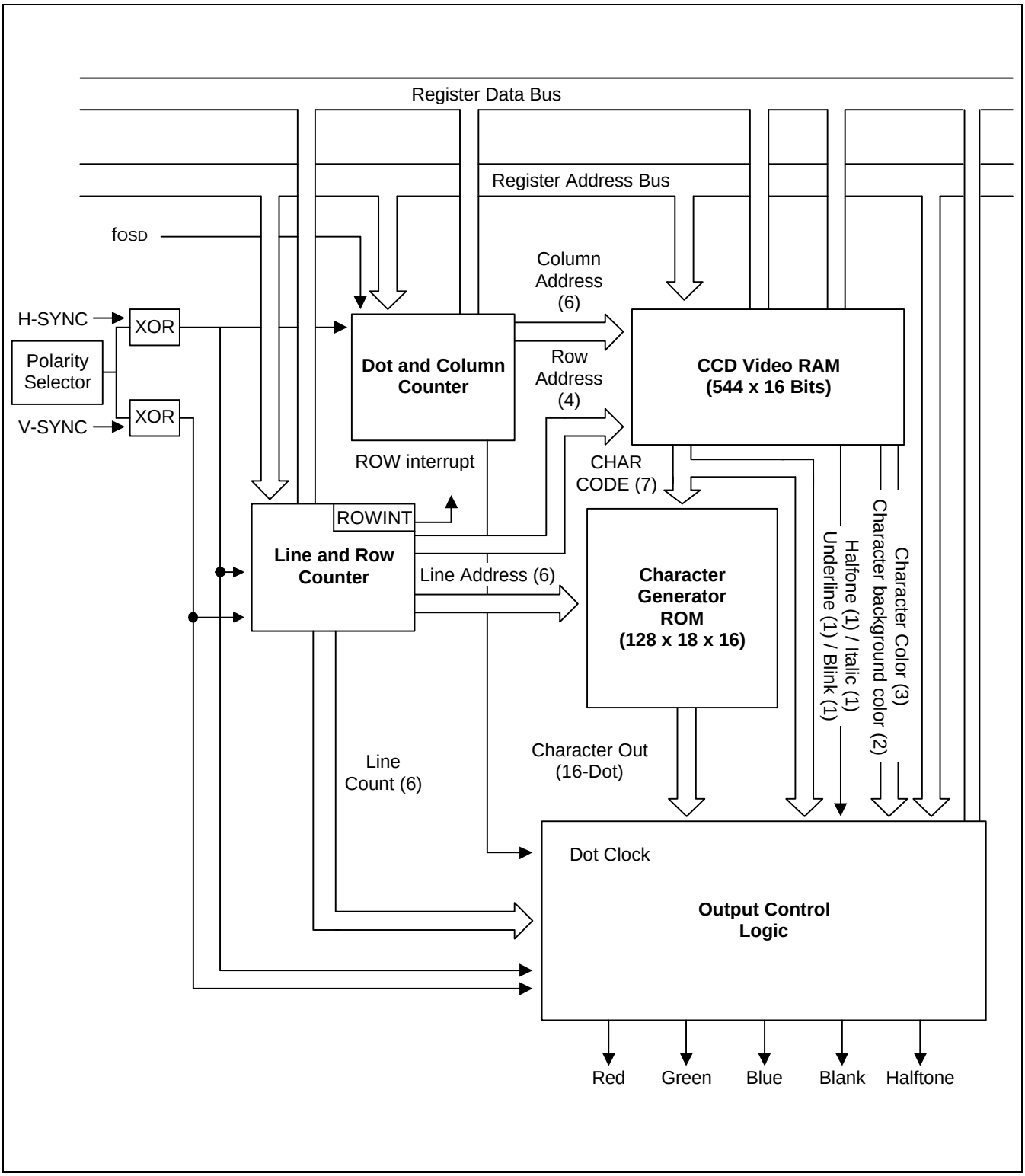


Figure 16-4. Closed Caption Display Function Block Diagram

OSD DISPLAY CHARACTER REGISTER

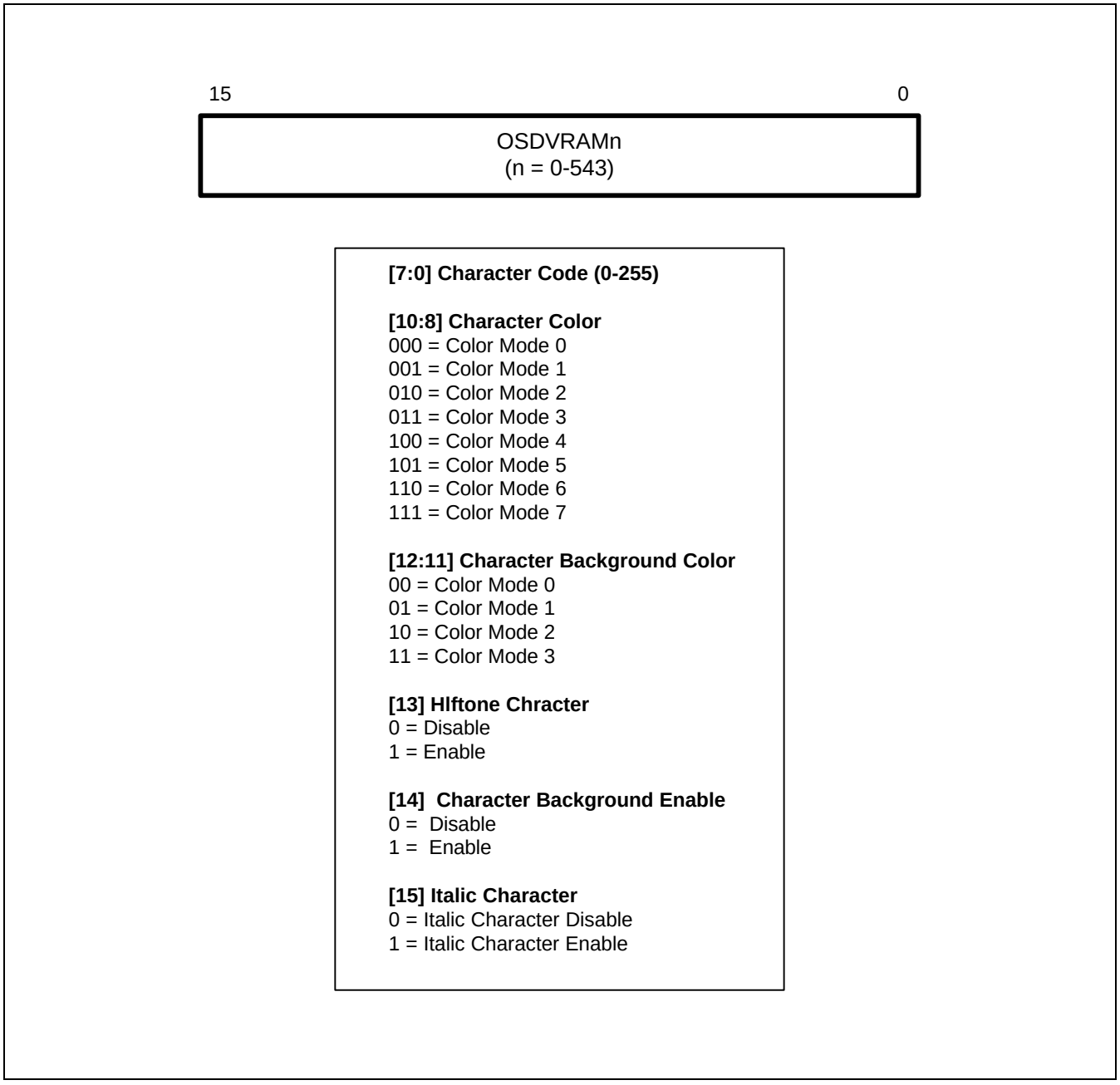


Figure 16-5. Video RAM Format (On-Screen Display Character)

CCD DISPLAY CHARACTER REGISTER

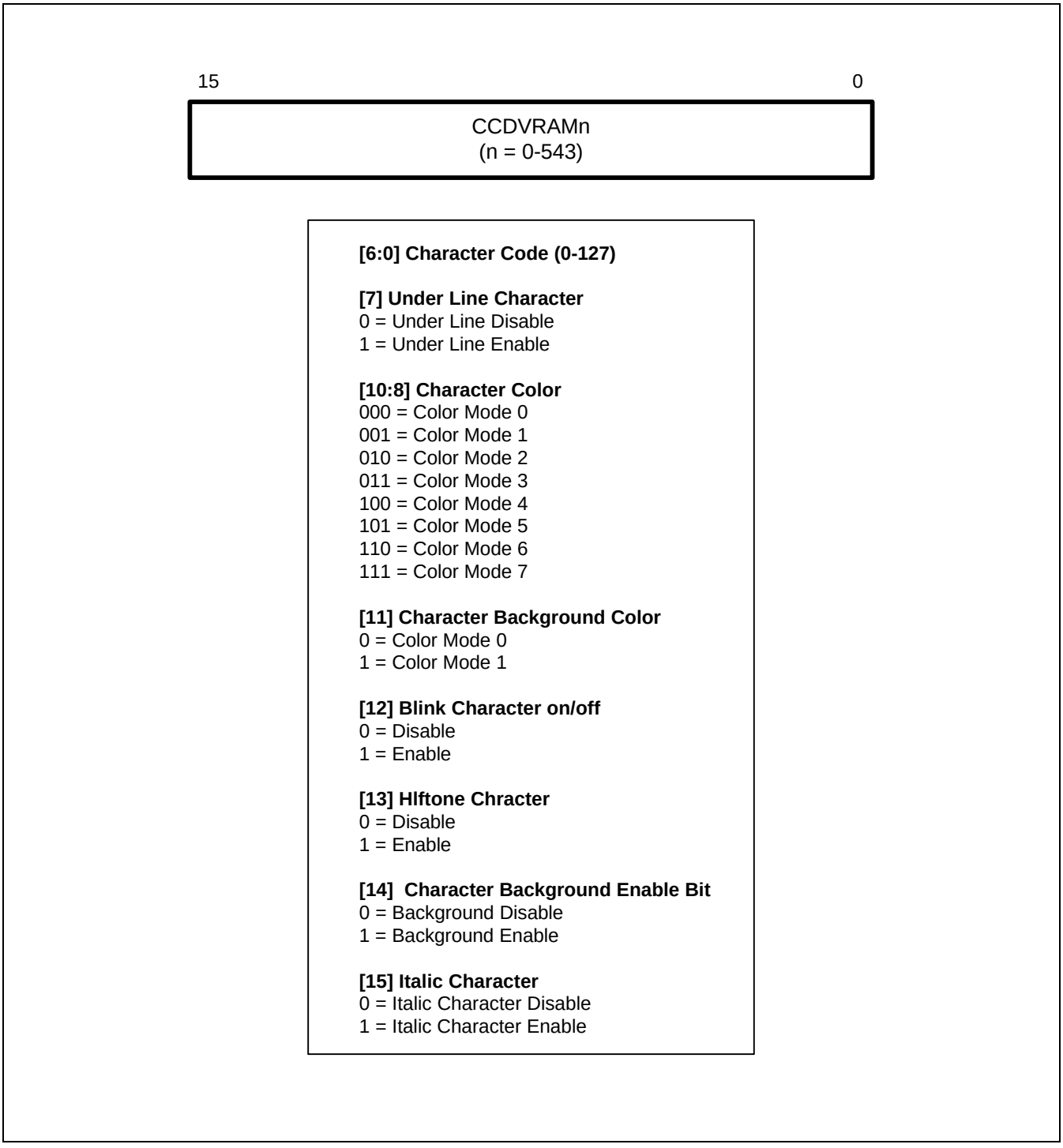


Figure 16-6. Video RAM Format (CCD Display Character)

OSD CONTROL REGISTER OVERVIEW

Twelve control registers are used to control specific functions of the on-screen display module:

There are Twelve control registers for OSD functions and one graphic data register:

OSDCON	OSD/CCD display control register
OSDPLTR	Palette color control register for Red
OSDPLTG	Palette color control register for Green
OSDPLTB	Palette color control register for Blue
OSDFLD	OSD Field control register
FDAECON	Fade control register
OSDBGD	OSD background control register
OSDVMGN	OSD margin control register
OSDCNT	OSD counter registers
OSGM 3/2/1	OSD graphic color control registers
OSGDATA	OSD graphic data register

These registers are described in this section within the context of the OSD hardware module description. For detailed quick-reference descriptions of the control register bit settings, please refer to Section 5, "Special Function Registers."

OSD/CCD DISPLAY CONTROL REGISTER (OSDCON)

The display control register OSDCON (0x30050), is used to enable and disable the on-screen display and CCD display, control horizontal and vertical character size, select polarity for OSD RGB, half tone and H/V sync, enable italic character, enable and disable V-sync interrupt, H-sync interrupt, OSD row interrupt, and line 21 interrupt, enable line graphic and graphic OSD and select OSD or CCD mode.

OSD/CCD Mode Selection (Bit 0)

You can select the OSD display mode and the CCD display mode by bit 0 of the OSDCON register.

Character Size Selection

The vertical character size is defined by bits 3 and 4 of the OSDCON register, the horizontal direction is defined by bits 1 and 2. There are four basic character size settings: × 1, × 2, × 3, and × 4. Size '× 1' is the smallest and '× 4' is the largest. For example, to display a '× 1' (horizontal) by '× 1' (vertical) size character, you should clear OSDCON.1-CHACON.4 to "0". To display a '× 4' by '× 4' size character, you should set bits 1-4 to '1111B'.

You can also combine different vertical and horizontal size selections to produce flattened or elongated characters. Character size selection can be possible in OSD mode. In CCD mode, character size is sticked 1 × 1 size.

Signal Polarity Selection

If bit 5 is set to "0", RGB color is output where OSD RGB output is in the high lend. If bit 5 is set to "1", RGB color is output in the low lend. If bit 6 is set to "0", halftone output is active in the high level. If bit 6 is set to "1" halftone output is active in the low level. If bit 7 is set to "0", the rising edge is selected to be the point recognizing the input of H-sync and V-sync. If bit 7 is set to "1", the falling edge is selected to be the point recognizing the input of H-sync and V-sync.

Italic Character Enable

Italic character is a lopsided form of characters requiring a blank character before and after it for a normal display. IF $f_{OSD,}$ $f_{CPU,}$ italic character display can be malfunction.

V-sync Interrupt Enable

Bit 9 enables or disables V-sync interrupts. A V-sync interrupt occurs when there is a V-sync input from outside. Either the rising edge (OSDCON.7 = "0") or falling edge (OSDCON.7 = "1") can be selected as the V-sync recognition point using OSDCON.7.

H-sync Interrupt Enable

Bit 10 enables or disables H-sync interrupts. An H-sync interrupt occurs when there is an H-sync input from outside. Either the rising edge (OSDCON.7 = "0") or the falling edge (OSDCON.7 = "1") can be selected as the H-sync recognition point using OSDCON.7.

OSD ROW INTERRUPT CONTROL

The S3C380D/F380D has a total of 16 OSD display rows. When enabled, an OSD ROW interrupt occurs in the first line of each row. Up to 16 OSD ROW interrupts can be generated, while this number can be reduced according to different settings in the top margin (OSDVMGN.9-.4), inter row space (OSDVMGN.3-.0), vertical character size (OSDCON.4-.3), and V-sync blank time (OSDVMGN.15-10). The ROW counter of OSDCNT.11-.8 informs the order of an OSD ROW interrupt occurring within a frame. The first ROW interrupt (OSDCNT.11-.8 = “0000B”) occurs at the point the top margin is completed. An OSD ROW interrupt is generated at the beginning of a ROW (for ROW 0 through ROW 15).

An OSD ROW interrupt allows different controls to each ROW. If the OSD control register is adjusted in the first OSD ROW interrupt (OSDCNT.11-.8 = 1111B) service routine affects the rows from ROW 0.

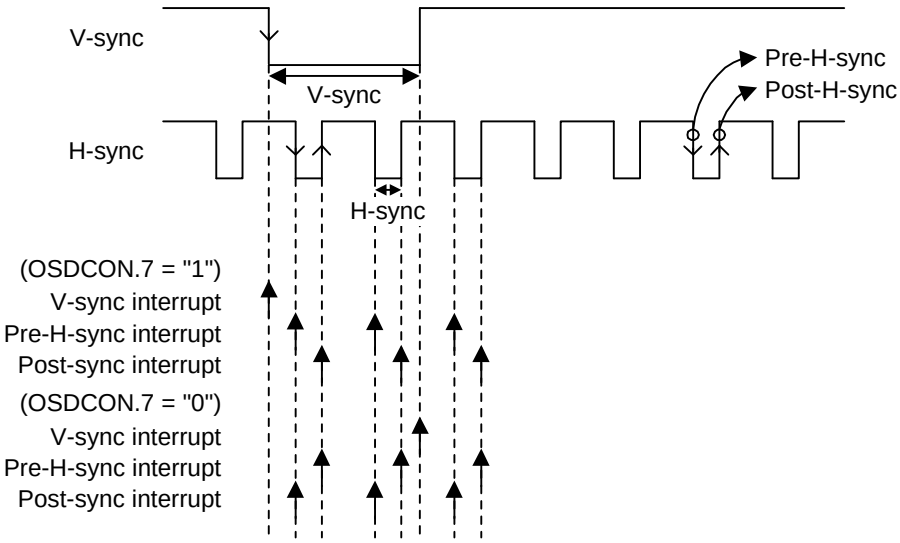
Graphic OSD Control

The S3C380D is capable of enabling or disabling each dot display comprising a font of 16 (horizontal dots) × 18 (vertical dots). It carry also control colors by two horizontal dots. This graphic OSD function is realized by using OSDCON.15-14, OSD graphic color data (OSGDATA), and OSD graphic color mode 3, 2, 1 register. There are two kinds of graphic display: display by font and by horizontal line. Display by font requires OSDCON. 15-14 to be set to “10B”. In this case, graphic display by line is disabled and the value “01H” should be written at the address of the video RAM for graphic display. In the font of the font ROM area but the content of OSGDATA is written inter location. As one font consists of 18 lines, there should be 18 updates of OSGDATA for a graphic data display. If OSDCON.15 = “0”, the 01H of video RAM displays the 01H font of the font ROM. For a line graphic display, OSDCON.15 should first be set to “1”. When OSDCON.14 is also “1”, OSGDATA is displayed.

OSD/CCD Display Enable

OSDCON.15 enables/disables R, G, B, and blank display. OSD/CCD display is also enabled (OSDCON.15 = “1”) or disabled (OSDCON.15 = “1”). OSDCON.0 selects either OSD mode or CCD mode (OSDCON.0 = “1”) for display. After a mode is selected, a video RAM format for OSD display or CCD display is chosen.

V and H-Sync Polarity Selection



15 0

OSDCON
0x30050, R/W

[0] OSD/CCD Mode
0 = OSD Mode
1 = CCD Mode

[2:1] Horizontal Character Size
00 = *1 Size
01 = *2 Size
10 = *3 Size
11 = *4 Size

[4:3] Vertical Characater Size
00 = *1 Size
01 = *2 Size
10 = *3 Size
11 = *4 Size

[5] OSD RGB Polarity
0 = Positive Signal Output
1 = Negative Signal Output

[6] Half Tone Polarity
0 = Active High Level
1 = Active Low Level

[7] V and Pre-H-sync Polarity (note)
0 = External Mode (Rising Edges)
1 = External Mode (Falling Edges)

[8] Italic Character
0 = Italic Character Disable
1 = Italic Character Enable

[9] V-sync Interrupt Enable Bit
0 = Disable V-sync Interrupt
1 = Enable V-sync Interrupt

[10] Pre-H-sync Interrupt Enable Bit
0 = Disable Pre-H-sync Interrupt
1 = Enable Pre-H-sync Interrupt

[11] OSD Row Interrupt Enable Bit
0 = Disable OSD Row Interrupt
1 = Enable OSD Row Interrupt

[12] Post-sync Interrupt Enable Bit
0 = Disable Post-H-sync Interrupt
1 = Enable Post-H-sync Interrupt

[13] Line Graphic Enable Bit
0 = Disable Line Graphic
1 = Enable Line Graphic

[14] Graphic OSD Enable Bit
0 = Disable Graphic OSD
(Video RAM (01H): Test Code)
1 = Enable Graphic OSD
(Video RAM (01H): Graphic Code)

[15] OSD/CCD Display Enable Bit
0 = Disable OSD/CCD
1 = Enable OSD/CCD

NOTE: Pre-H-sync means 1st edge H-sync.
Post-H-sync edge polarity is selected by inverting edge of pre-H-sync.
Bit7 of OSDCON select pre-H-sync polarity, so post-H-sync polarity is inversion edge of pre H-sync

Figure 16-7. OSD Display Control Register (OSDCON)

CHRACTER SIZE

OSD Character

An OSD character consists of 16 × 18 (width × height) dots. One TV frame comprises odd and even fields, displaying one line. That is, the even and the add fields display the sames line simultaneously.

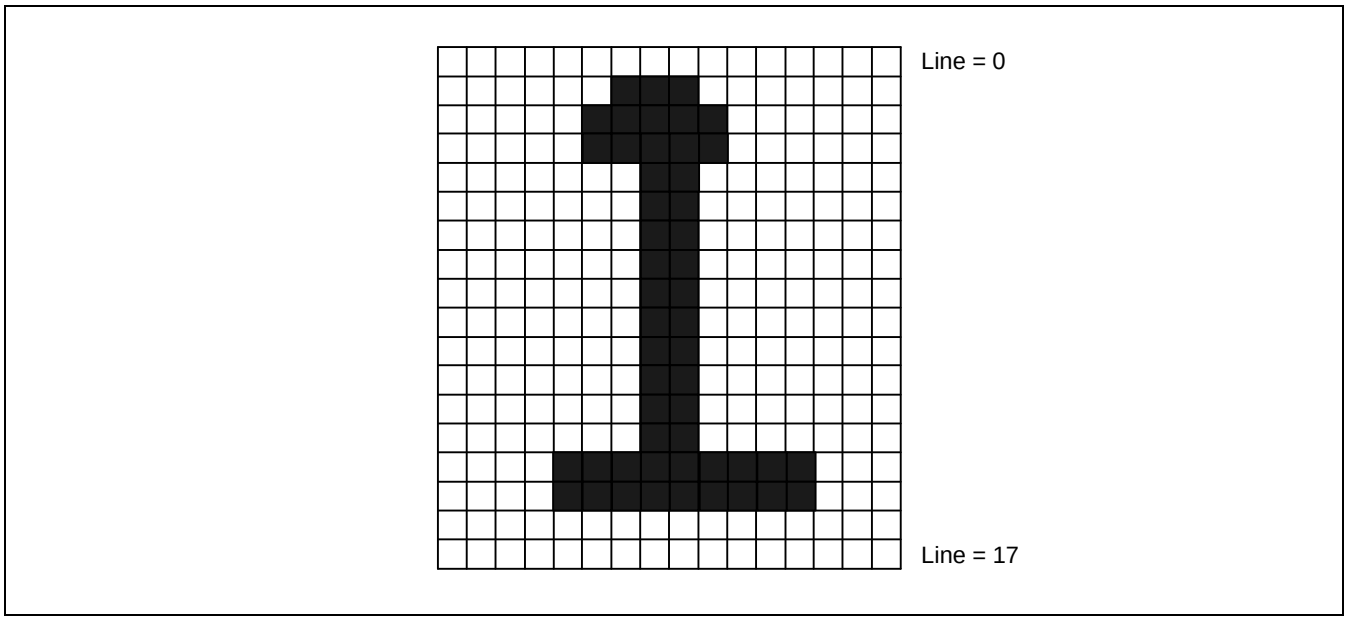


Figure 16-8. OSD Character

CCD Character

A CCD character consists of 8 × 13 (width × height) dots. But in real, a character is composed of 8 × 9 dots. Line 10 is used for underlining function, and line 9, 11, and have no use. In CCD character display, one TV frame consists of odd and even fields. The odd field displays odd lines while the even field displays even lines. As the display size is smaller than in OSD display mode, OSD clock (f_{OSD}) must be adjusted to modify the horizontal size.

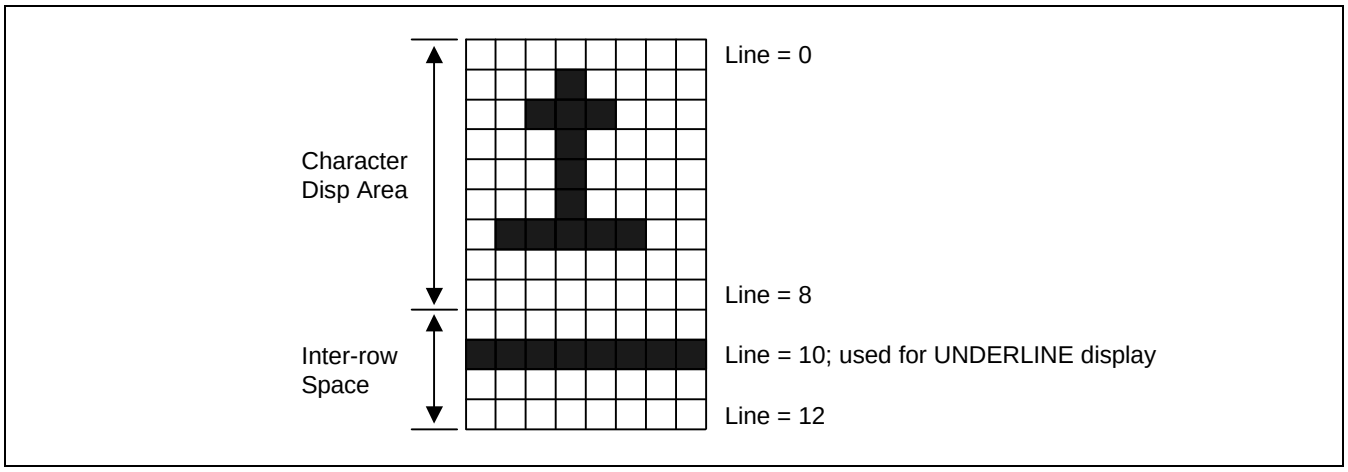


Figure 16-9. CCD Character

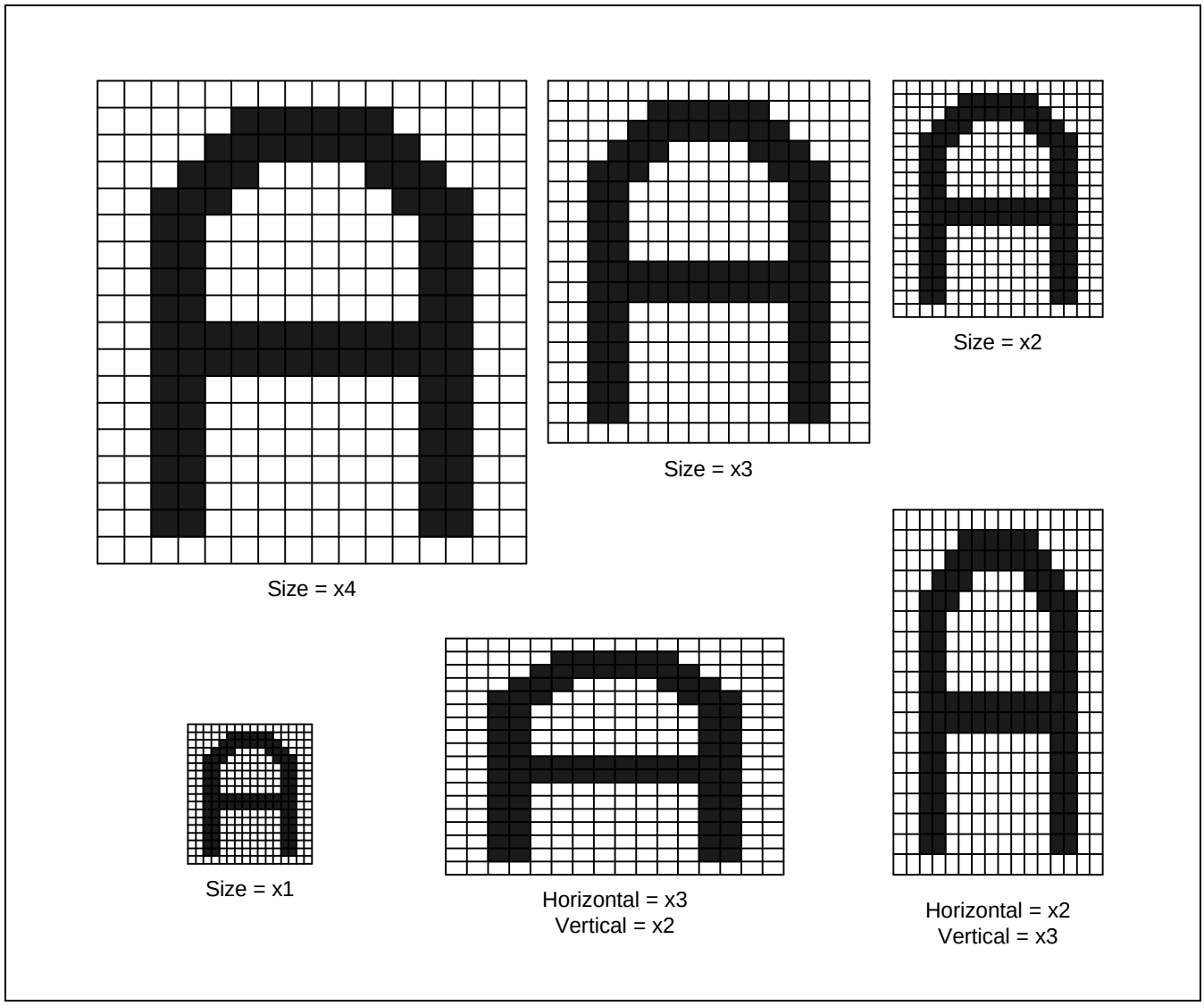


Figure 16-10. OSD Character Sizing Dimensions

FADE-IN AND FADE-OUT CONTROL REGISTER (FADECON)

The OSD block lets you program fade-in and fade-out displays. A *fade-in display* is one in which a character matrix is displayed incrementally until the complete character "appears". A *fade-out display* shows the complete character matrix first and then decrements the matrix line-by-line until the character "disappears" from the display field.

The address of the character display (and the specific line) to be faded-in or faded-out is selected by writing bit values into the FADECON register. Bits 11–8 in the FADECON register specify the 4-bit video RAM address of one of the sixteen rows (0–11) of the fade display. Bits 0–5 in the FADECON register specify the 6-bit character line address (0–17) and inter row space line address (18-32) within the selected row.

Fade direction is controlled by FADECON.6. There are two choices of fade direction: before (FADECON.6 = "0") and after (FADECON.5 = "1"). When you select *fade before*, the character matrix is faded starting with line 0. When you select *fade after*, the matrix is faded starting with line 32, inter row space line 15 (The line 32 start position is only a suggestion, however, as the fade interval is assignable by software.) To enable the fade function, you should set FADECON.7 to "1". (FADECON.12-15 is not used).

NOTE

To avoid confusion in determining fade row and line addresses in the FADECON register, please note that a *line* is a horizontal value that encompasses the entire character display field while a *row* is a horizontal value for the character display matrix.

[5:0] Fade Line Address (0-17h) + Inter ROW Space (15H)
000000 = Line 0
000001 = Line 1
000010 = Line 2
000011 = Line 3
000100 = Line 4
000101 = Line 5
000110 = Line 6
000111 = Line 7
001000 = Line 8
001001 = Line 9
001010 = Line 10
001011 = Line 11
001100 = Line 12
001101 = Line 13
001110 = Line 14
001111 = Line 15
010000 = Line 16
010001 = Line 17
010010 = Inter Row Space Line 1
010011 = Inter Row Space Line 2
010100 = Inter Row Space Line 3
010101 = Inter Row Space Line 4
010110 = Inter Row Space Line 5
010111 = Inter Row Space Line 6
011000 = Inter Row Space Line 7
011001 = Inter Row Space Line 8
011010 = Inter Row Space Line 9
011011 = Inter Row Space Line 10
011100 = Inter Row Space Line 11
011101 = Inter Row Space Line 12
011110 = Inter Row Space Line 13
011111 = Inter Row Space Line 14
100000 = Inter Row Space Line 15
100001 = Not used
111111 = Not used

[6] Fade Direction Selection Bit
0 = Fade Before Matrix
1 = Fade After Matrix

[7] Fade Function Enable Bit
0 = Fade Disable
1 = Fade Enable

[11:8] Fade Row Address Selection
0000 = Row 0
0001 = Row 1
0010 = Row 2
0011 = Row 3
0100 = Row 4
0101 = Row 5
0110 = Row 6
0111 = Row 7
1000 = Row 8
1001 = Row 9
1010 = Row 10
1011 = Row 11
1100 = Row 12
1101 = Row 13
1110 = Row 14
1111 = Row 15

[15:12] No effect

Figure 16-11. OSD Fade Control Register (FADECON)

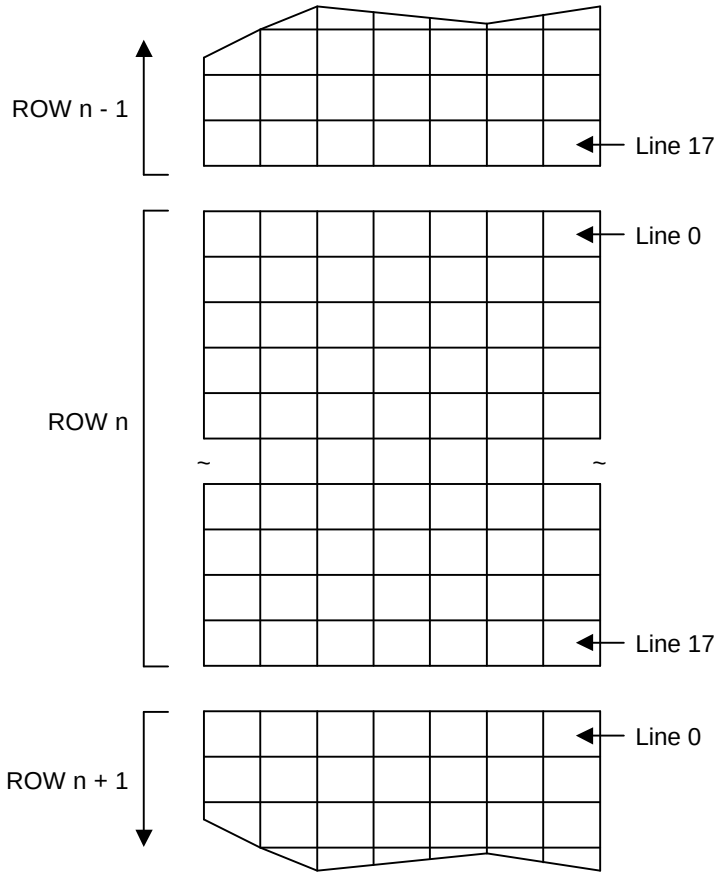


Figure 16-12. Line and Row Addressing Conventions

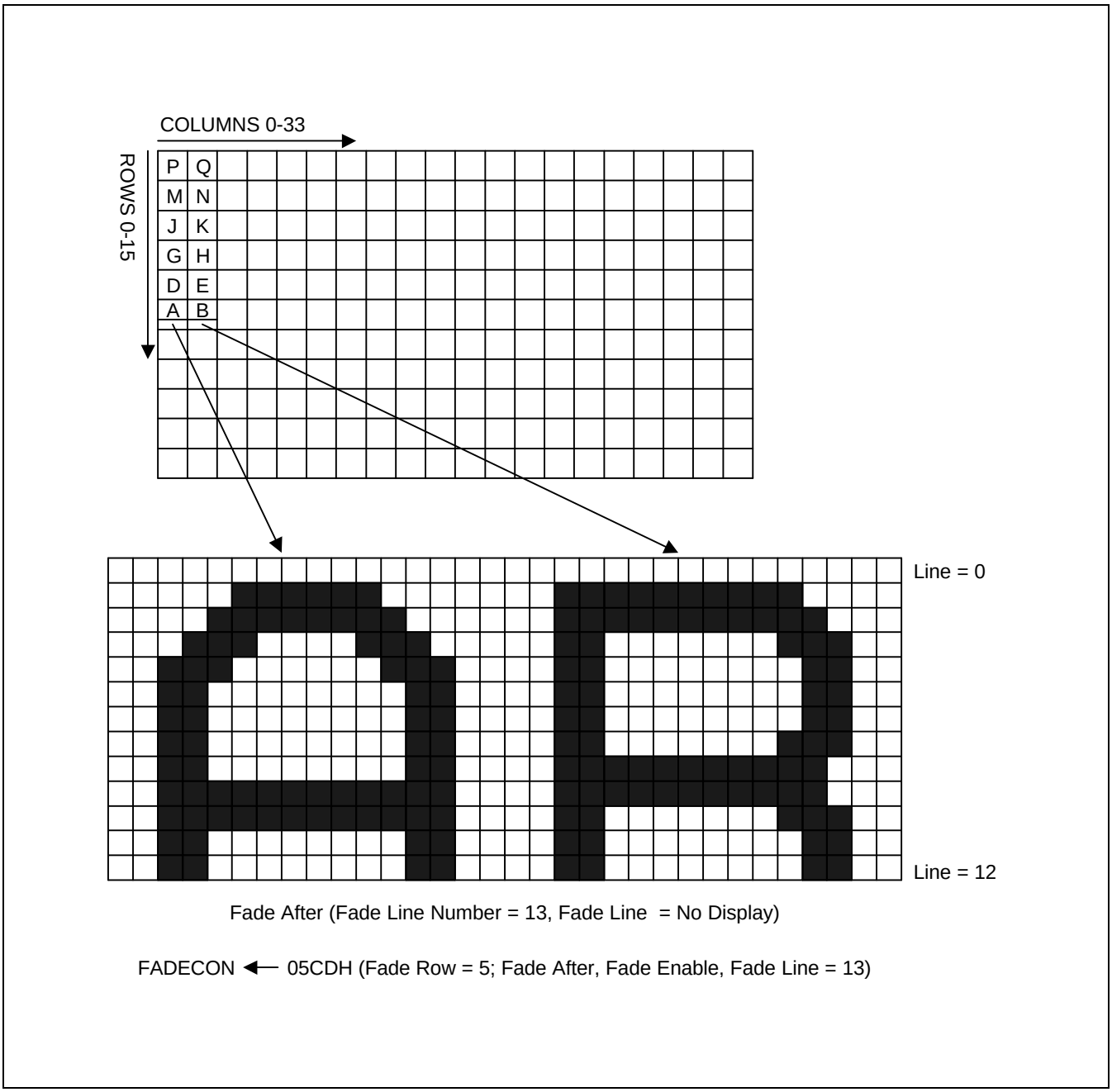


Figure 16-13. OSD Fade Function Example: Fade After

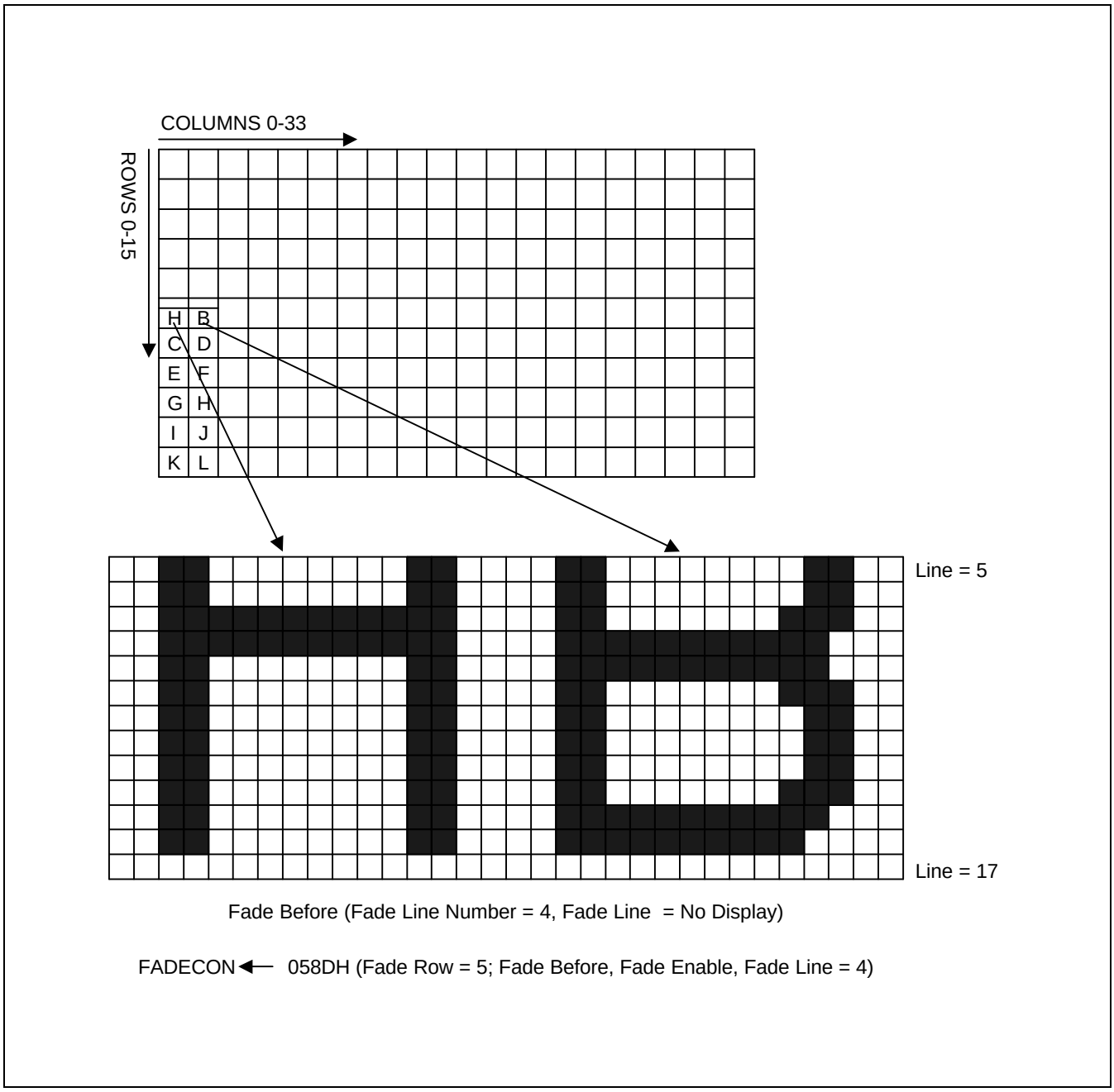


Figure 16-14. OSD Fade Function Example: Fade Before

OSD BACKGROUND CONTROL REGISTER (OSDBGD)

OSD Frame Background Control

OSDBGD.3 controls enabling/disabling OSD frame background display. If it is set to “1”, the color mode defined by OSDBGD.0-3 is selected as that of OSD palette color mode R, G, B. For instance, if color mode 3 9OSDBGD.0-3 = “1011B”) is chosen, bit 6-7 (OSDPLTR.6-7, OSDPLTG.6-7, OSDPLTB.6-7) of OSD palette color mode register R, G, B, which are designed for color mode 3 defines the background color.

When OSDBGD.3 = “0”, frame background is disabled regardless of the values of OSDBGD.0-2. OSDBGD.4 enables or disables the halftone function of the frame background area. When it is set to “1”, the entire area, excluding the parts of characters, is disabled as halftone. If the frame background color is displayed by OSDBGD.0-3, the frame background color is maintained.

Inter-Row Space Control

OSDBGD.5 defines the color of the inter-row space with a help of character background (DSPOSDn.14 or DSPCCDn.14; n = 0-543) or frame background color enable bit (OSDBGD.3). When set to “0”, the inter-row space color is the save as the character background color. When set to “1”, the color is the same as the frame background color.

OSDBGD.6 defines the halftone function of the inter-row space area, which depends on the halftone enable/disable state of the character background or the frame background areas. If OSDBGD.6 = “0”, the halftone function of the inter-row space is in the same state as the halftone function in the character background area. If it is “1”, the function follows the save pattern as that of the frame background area.

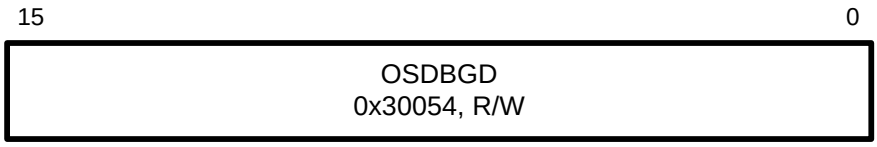
Note that when the halftone function of the frame background area is enabled (OSDBGD.3 = “1”), the function is enabled for the inter-row space, regardless of the value of OSDBGD.6.

Inter-Column Space Control

OSDBGD.8-10 control inter-column space, which ranges from 0 to 7 OSD clock (f_{OSD}).

Left Margin Control

Left margin refers to the space between the far left and the first character displayed on the screen. Defined by OSDBGD.11-15, this margin is in the range of 16 OSD clock (f_{OSD}) to 47 OSD clock.



[2:0] Frame Background Color
000 = Color Mode 0
001 = Color Mode 1
010 = Color Mode 2
011 = Color Mode 3
100 = Color Mode 4
101 = Color Mode 5
110 = Color Mode 6
111 = Color Mode 7

[3] Frame Background
0 = Disable
1 = Enable

[4] All Frame H/T Function (1)
0 = Disable
1 = Enable

[5] Inter-row Space Color
0 = Character background color
1 = Frame background color

[6] Inter-row Space Half tone (Only when All frame HT = 0)
0 = Depend on character background
1 = Depend on frame background

[7] No effect

[10:8] Inter-Column Spacing Control
000 = No Inter-column Spacing
001 = Inter-Column Spacing = 1 OSD Clock
010 = Inter-Column Spacing = 2 OSD Clock
011 = Inter-Column Spacing = 3 OSD Clock
100 = Inter-Column Spacing = 4 OSD Clock
101 = Inter-Column Spacing = 5 OSD Clock
110 = Inter-Column Spacing = 6 OSD Clock
111 = Inter-Column Spacing = 7 OSD Clock

[15:11] Left Margin Display Position Control (16 + (0-32 dot)
00000 = Left Margin (16 OSD Clocks)
00001 = Left Margin (16 + 2 OSD Clocks)
00010 = Left Margin (16 + 4 OSD Clocks)
.
.
.
11110 = Left Margin (16 + 62 OSD Clocks)
11111 = Left Margin (16 + 64 OSD Clocks)

NOTE: When all frame HT = 1, the inter-row space interval is HT enable even if inter-row space HT = 0 (OSDBGD.6 = 0).

Figure 16-15. OSD Background Color Control Register

BACKGROUND COLOR CONTROL

The background color control register, OSDBGD, lets you select background colors for both the display frame and characters:

— *Frame background* is the full-screen display field upon which the character display is imposed.

— *Character background* is a color field surrounding an individual character. To enhance readability, the background is usually a color that contrasts or highlights the characters in a pleasing manner.

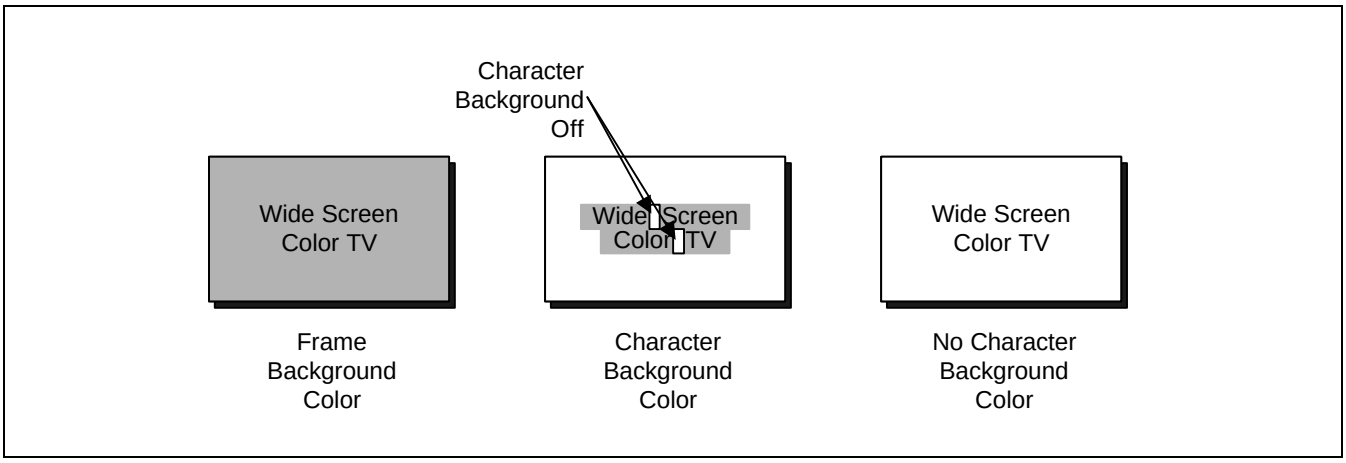


Figure 16-16. Background Color Display Conventions

Background color and Halftone Function Mode

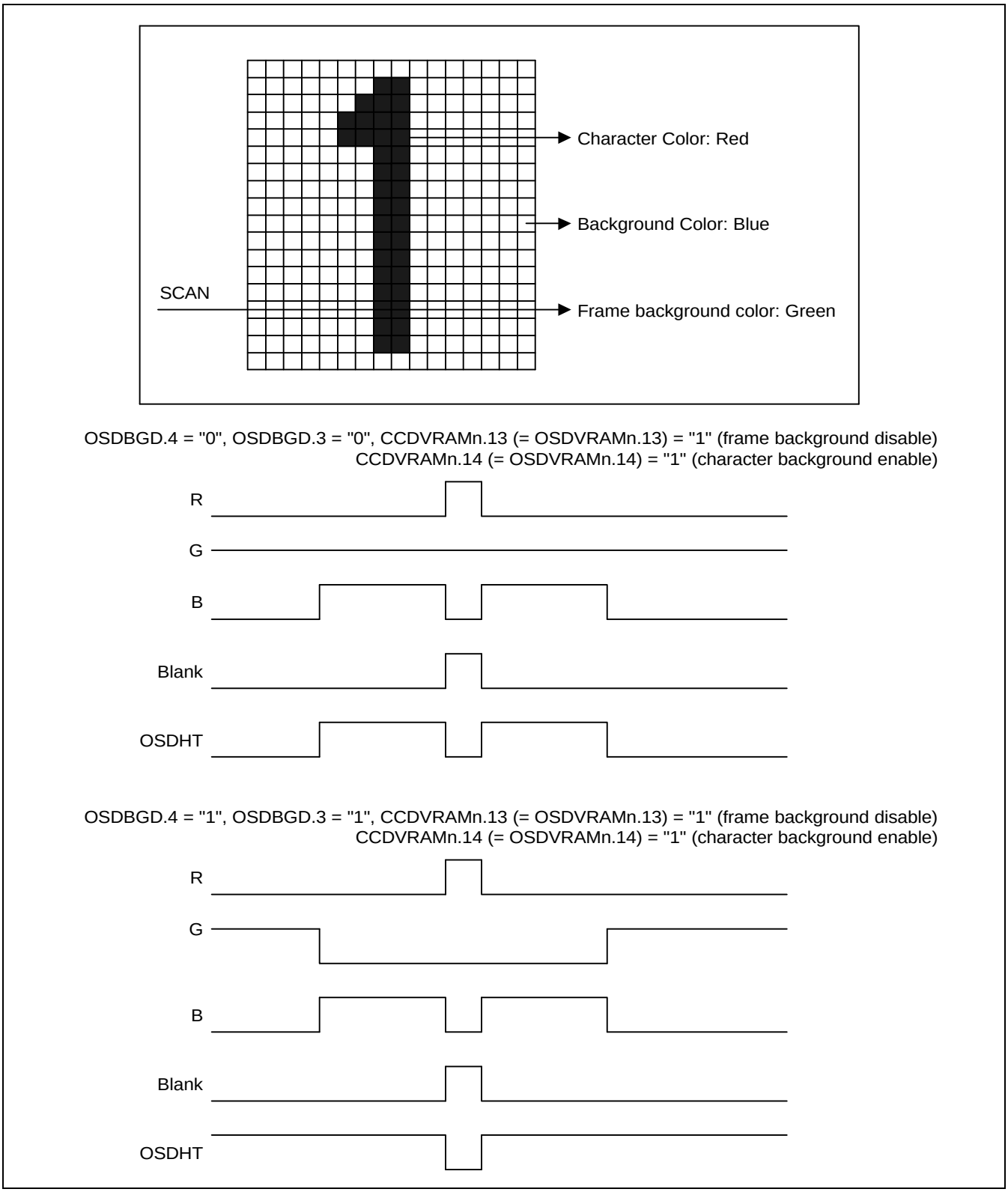


Figure 16-17. Halftone or Character Background Signal Output

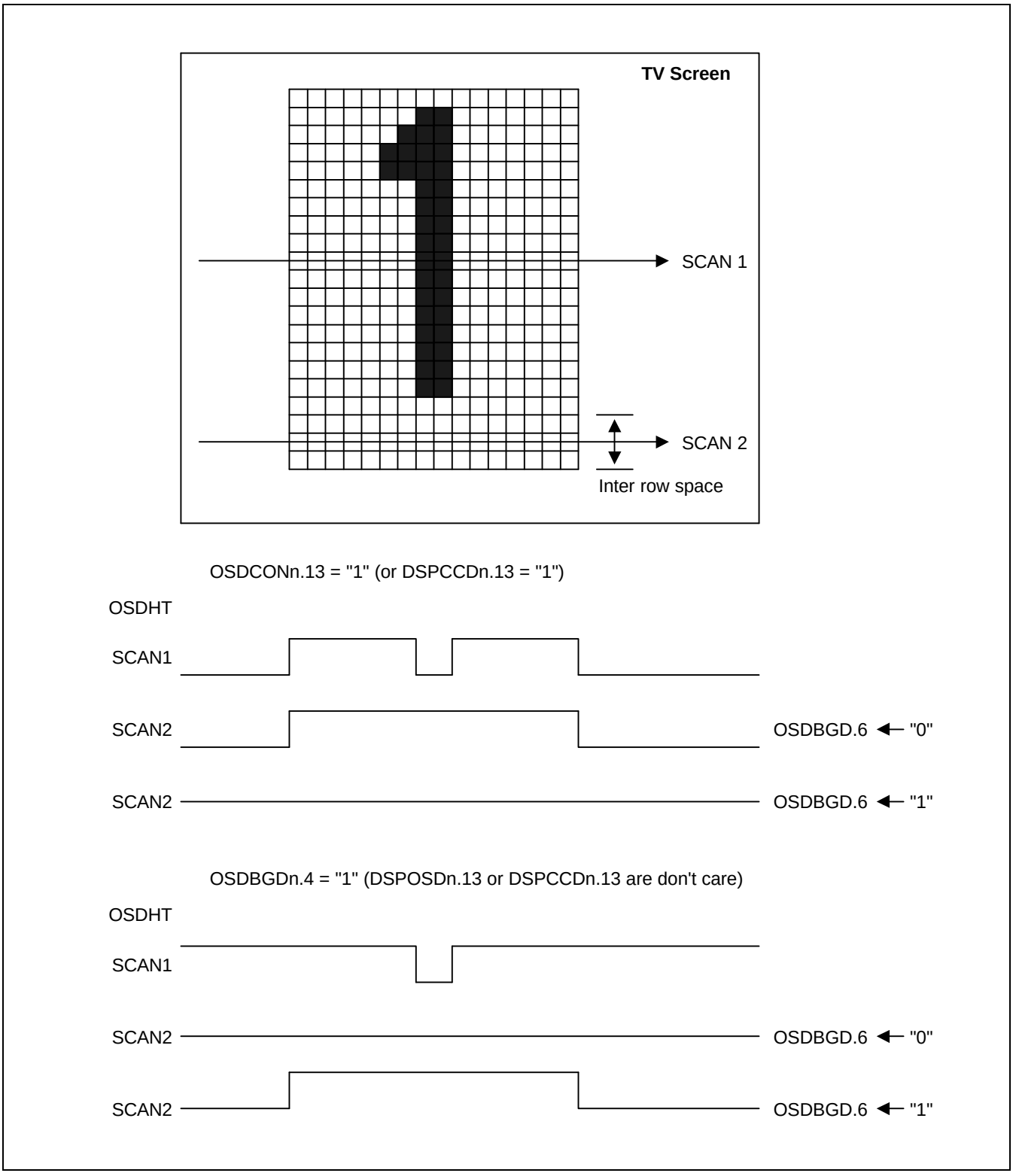


Figure 16-18. Halftone Control Signal Output Options

OSD MARGIN REGISTER (OSDVMGN)

The OSD margin control register, OSDVMGN, controls the top margin, V-sync blank time, and inter-row spacing. *Top margin* is the distance (in H-sync pulses) to the top row of a character display from the top edge of its display frame. *Inter-row spacing* is the distance (in H-sync pulses) between two rows of displayed characters. You can select different inter-row space to every row by writing a value in a row interrupt.

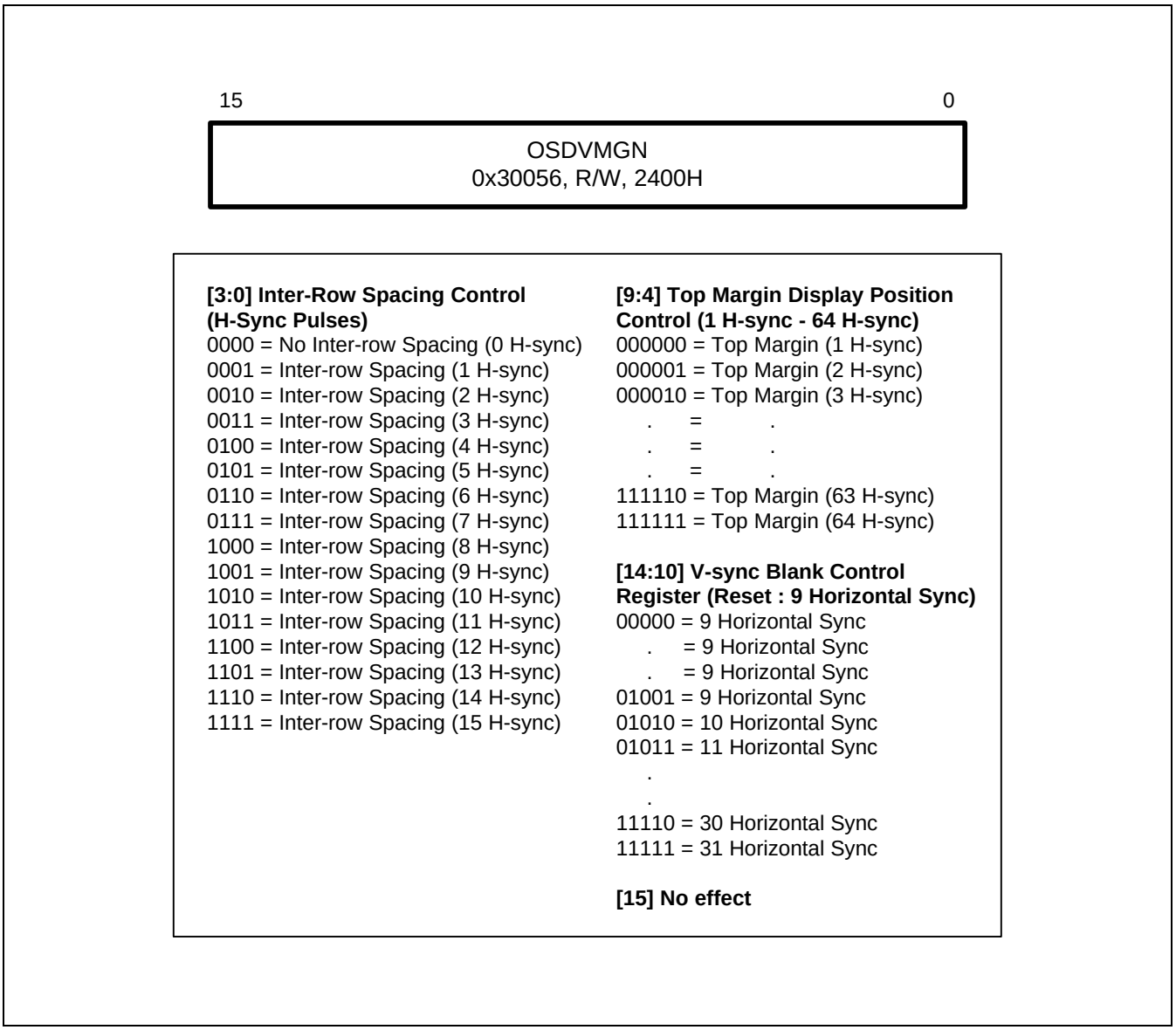


Figure 16-19. OSD Top Margin Control Register (OSDVMGN)

Upon accepting external V-sync, V-sync blank control bit allows an extension of the sync area after detecting the V-sync edge. The extension is made in terms of the number of horizontal sync. A minimum of 9 horizontal sync can be extended upto 31 by adjusting the value in OSDVMGN.10-14.

V-sync Blank and Top Margin Timing Diagram

The following is a timing diagram simplified with external V-sync input of H-sync input signals. V-sync blank time and top margin are controlled by OSDVMGN. OSDVMGN.10-14 ← 01011B (V-sync blank time = 11 Horizontal sync). OSDVMGN.4-9 ← 010000B (top margin 16).

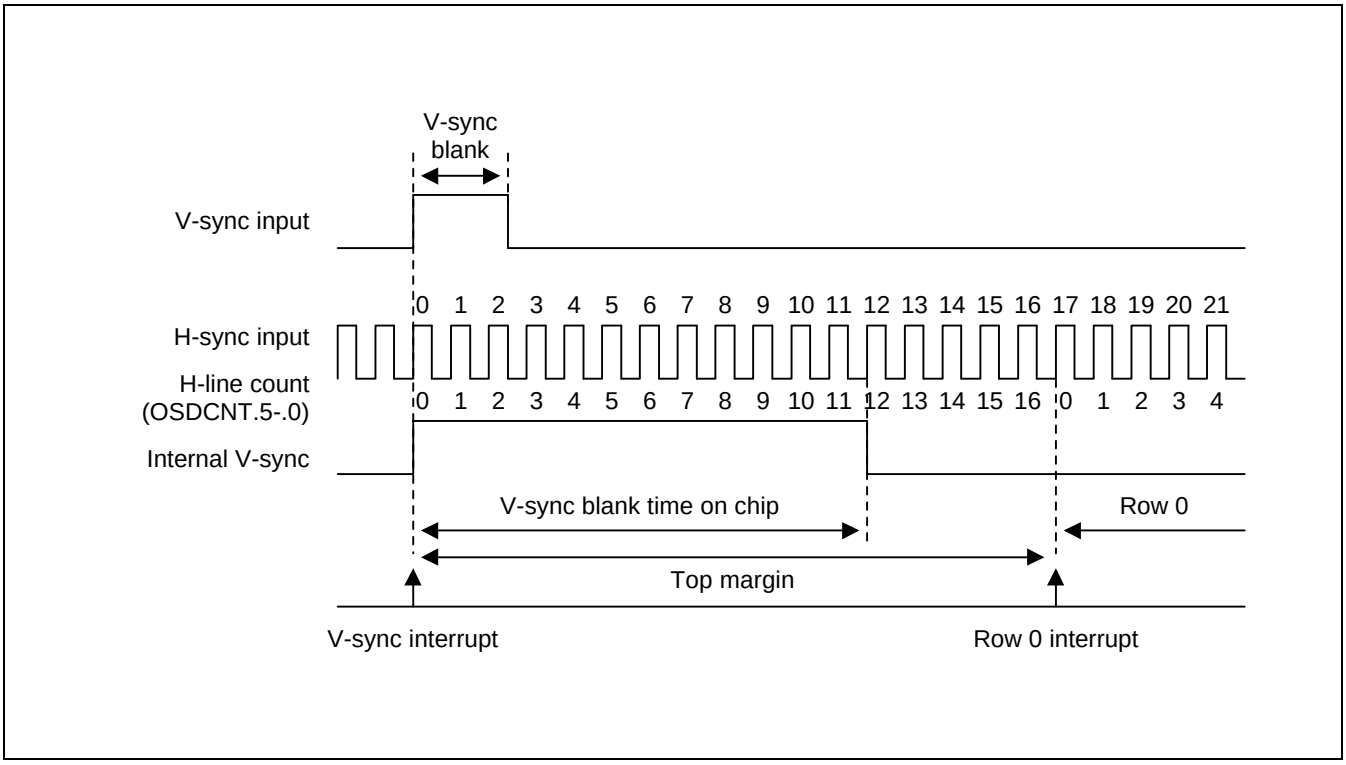


Figure 16-20. V-sync Blank and Top Margin Timing Diagram

Calculation Margin

1. Top margin (OSDVMGN.4-.9) < V-sync blank time (OSDVMGN.10-.14)
   Top margin = 64 – V-sync blank time (OSDVMGN.10-.14) + OSDVMG.4-.9 + 1
   For example, OSDVMGN.4-.9 = 5, OSDVMGN.10-.14 = 10
   Top margin = 64 – 10 + 5 + 1 = 60

2. Top margin (OSDVMGN.4-.9) ≥ V-sync blank time (OSDVMGN.10-.14)
   Top margin = OSDVMGN.4-.0 – V-sync blank time (OSDVMGN.10-.14) + 1
   For example
   1) OSDVMGN.4-.9 = 16, OSDVMGN.10-.14 = 11
   Top margin = 16 – 11 + 1 = 6
   2) OSDVMGN.4-.9 = 9, OSDVMGN.10-.14 = 9
   Top margin = 9 – 9 + 1 = 1

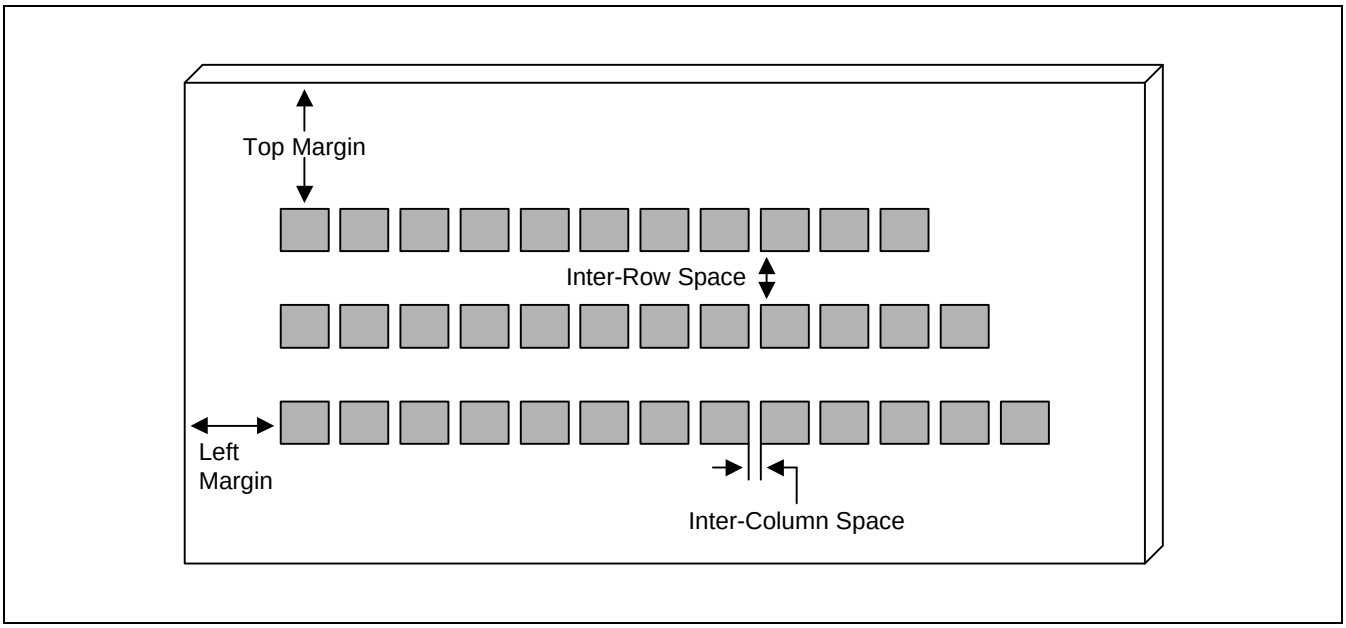


Figure 16-21. OSD Display Formatting and Spacing Conventions

Calculating Row and Column Spacing

Inter-row spacing and inter-column spacing are controlled by the OSDVMGN and OSDBGD registers. You can select from zero to seven dots for inter-column spacing and OSDVMGN from zero to fifteen dots for inter-row spacing.

For inter-row spacing, the desired spacing value (0–7) is written to bits 0–3 of the OSDVMGN register. For inter-column spacing, the desired spacing value (0–15) is written to bits 8–10 of the OSDBGD register.

Calculating Margin Settings

By writing a value to OSDVMGN.4– OSDVMGN.9, you can set the top margin at the top margin dot value + 1 (TMG). Because TMG is a 6-bit value, you can select any dot value in the range 1–64. For detail description, see page 16-25.

By writing a value to OSDBGD.11–OSDBGD.15, you can set the left margin at 16 + the left margin dot value (LMG). Because LMG is a 5-bit value, you can select any dot value in the range 0–31. The zero position for the left margin is always 16 dots.

— Top margin: see page 16-25

— Left margin = 16 + (left margin register value) dot clock

— Inter-column space = (Register value) dot clock

— Inter-row space = (Register value) H

OSD PALETTE COLOR MODE REGISTERS (OSDPLTR, OSDPLTG, OSDPLTB)

OSD palette color mode register R, G, B controls the color of OSD R, G, B output. OSDPLTR, OSDPLTG, and OSDPLTB are composed of 16 bits each, in which the combinations of bit 0-bit 1, bit 2-bit 3, bit 4-bit 5, bit 6-bit 7, bit 8-bit 9, bit 10-bit 11, bit 12-bit 13, and bit 14-bit 15 define color mode 0 to 7, respectively.

Each color mode can express upto bit color by combing the three registers (OSDPLTR, OSDPLTG, OSDPLTB). As one color mode can select one color out of 64 choices, and there are 8 color modes, a total of 8 colors can be displayed at time. For example, when combining color mode 0, each of OSDPLTR.0-1, OSDPLTG.0-1, and OSDPLTB.0-1 can produce 4 kids of red level, which can be multiplied upto 64 combinations.

Each 2 bits define a color mode make 4 color levels available. When the standard lighting is 100 %, the value “00B” means disabled, “01B” means 33 % and “10B” means 66 % of the standard light level.

In CCD display mode, the color mode defined by video RAM bit 8-10, and bit 11 represents the colors set by OSDPLTR, OSDPLTG, and OSDPLTB. In OSD display mode, the color mode defined by video RAM bit 8-10 and bit 11-12 and OSDBGD.0-2 represent them.

[1:0] OSD Mode 0 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[3:2] OSD Mode 1 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[5:4] OSD Mode 2 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[7:6] OSD Mode 3 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[9:8] OSD Mode 4 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[11:10] OSD Mode 5 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[13:12] OSD Mode 6 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[15:14] OSD Mode 7 Red Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

Figure 16-22. OSD Palette Color Mode R

[1:0] OSD Mode 0 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[3:2] OSD Mode 1 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[5:4] OSD Mode 2 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[7:6] OSD Mode 3 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[9:8] OSD Mode 4 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[11:10] OSD Mode 5 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[13:12] OSD Mode 6 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[15:14] OSD Mode 7 Green Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

Figure 16-23. OSD Palette Color Mode G

[1:0] OSD Mode 0 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[3:2] OSD Mode 1 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[5:4] OSD Mode 2 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[7:6] OSD Mode 3 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[9:8] OSD Mode 4 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[11:10] OSD Mode 5 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[13:12] OSD Mode 6 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

[15:14] OSD Mode 7 Blue Level
00 = Disable
01 = 33 %
10 = 66 %
11 = 100 %

Figure 16-24. OSD Palette Color Mode B

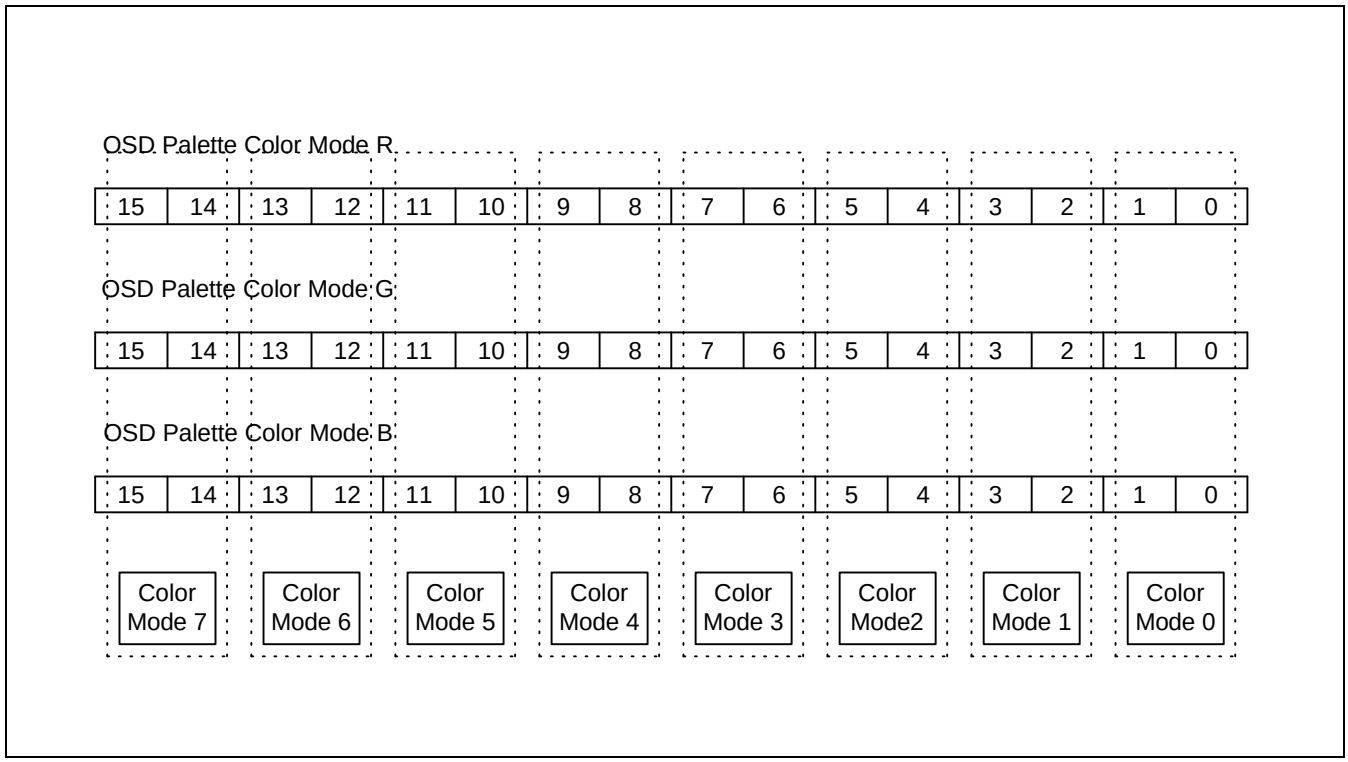


Figure 16-25. OSD Palette Color Mode R, G, B

When control register set as
OSDPLTR = 1100 1100 1010 0101B
OSDPLTG = 1111 0000 1110 1001B
OSDPLTB = 1110 0100 1110 1110B

The color of each color mode
Color mode 0 = 33% red level (OSDPLTR.1-.0) + 33% green level (OSDPLTG.1-.0)
+ 66% blue level (OSDPLTB.1-.0)
Color mode 1 = 33% red level (OSDPLTR.3-.2) + 66% green level (OSDPLTG.3-.2)
+ 100% blue level (OSDPLT3.-.2)

OSD GRAPHIC DISPLAY

Registers OSDCON, OSGDATA, OSGM3, OSGM2, and OSGM1 are used for OSD graphic display. OSD graphic display can be achieved by two methods. One is to select a fixed area and displaying it in units of 1 font area (OSDCON.14 = “1”, OSDCON.13 = “0”), and the other is to display any area that you want (OSDCON.14-.13 = “11”).

Fixed Area Display

OSDCON.15 = 1 (OSD display enable)
OSDCON.14-.13 = “10” (graphic OSD enable, line graphic disable)
OSGDATA = 1111 0000 1111 0000B
OSGM1 = 1111 1111 0000 0000B
OSGM2 = 1111 0000 1111 0000B
OSGM3 = 1100 1100 1100 1100B

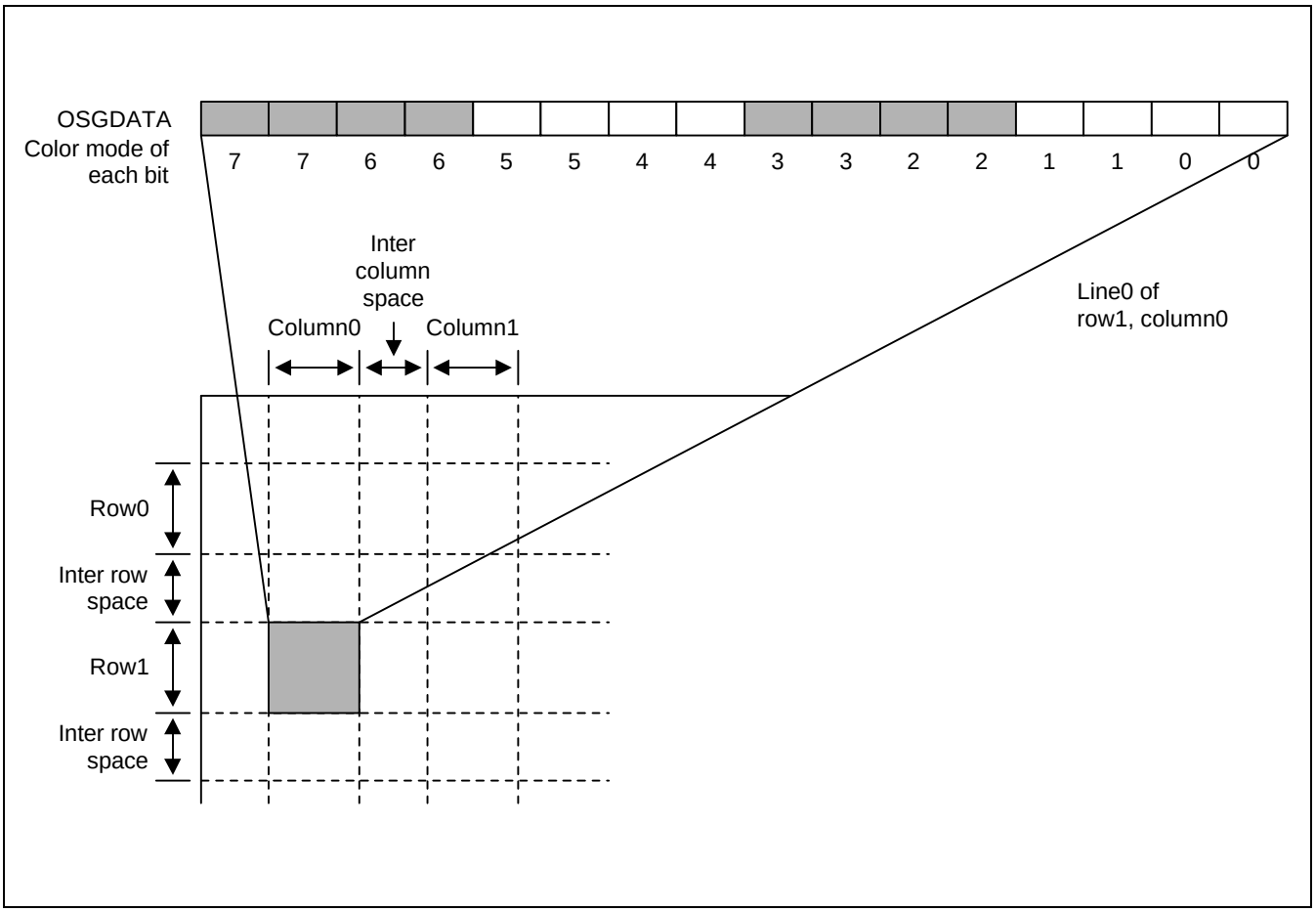


Figure 16-26. Graphic Data Display

To display a graphic on row1, column0 as shown in the figure above, you must write “01H” on bits 0-7 of video RAM address 22H (OSDCON.0 = “0”; in OSD mode), and write the OSGDATA value before OSDCNT’s line counter comes to row1, column0. At this time, the color of each bit is determined by the color mode selected by OSGM1, OSGM2, and OSGM3. The color of each color mode is the one determined by OSDPLTR, OSDPLTG, and OSDPLTB. Row1’s inter-row space area is displayed as graphic data, and the inter-column space area is not displayed.

The area set to “1” by OSGDATA in Figure 16-26 displays the color determined by each color mode, and the area set to “0” by OSGDATA is not displayed. If the character background is enabled (by OSDCON and video RAM), the area set to “0” by OSGDATA shows the color determined by OSGM1, OSGM2, and OSGM3’s color mode, and the color set as the character background color is displayed in the inter-column space area. If you want to display a graphic image from row1, column0’s lines 0-17, you must update the OSGDATA for each line’s H-sync input.

Display without Fixed Location

In this case, you need to enable the line graphic (OSDCOM.13 = “1”) while following the same steps as fixed area display). When the line graphic is enabled, the current graphic data (OSGDATA) and the contents of OSGM1, OSGM2, and OSGM3 and displayed without regard to if “0” is written video RAM’s bits 0-7. The inter-row space area also remains undisplayed in this case.

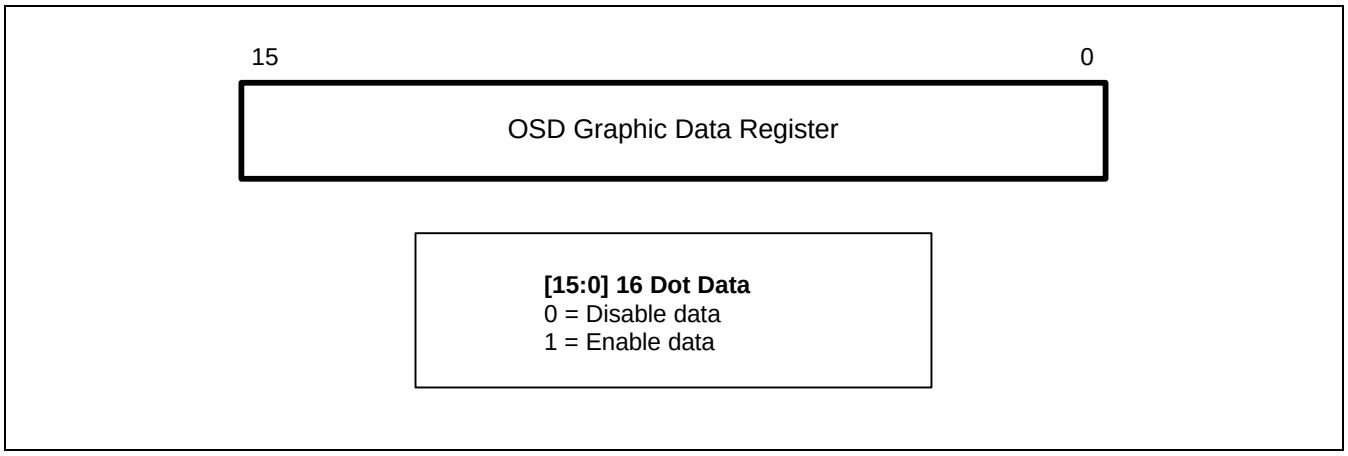


Figure 16-27. OSD Graphic Data Register

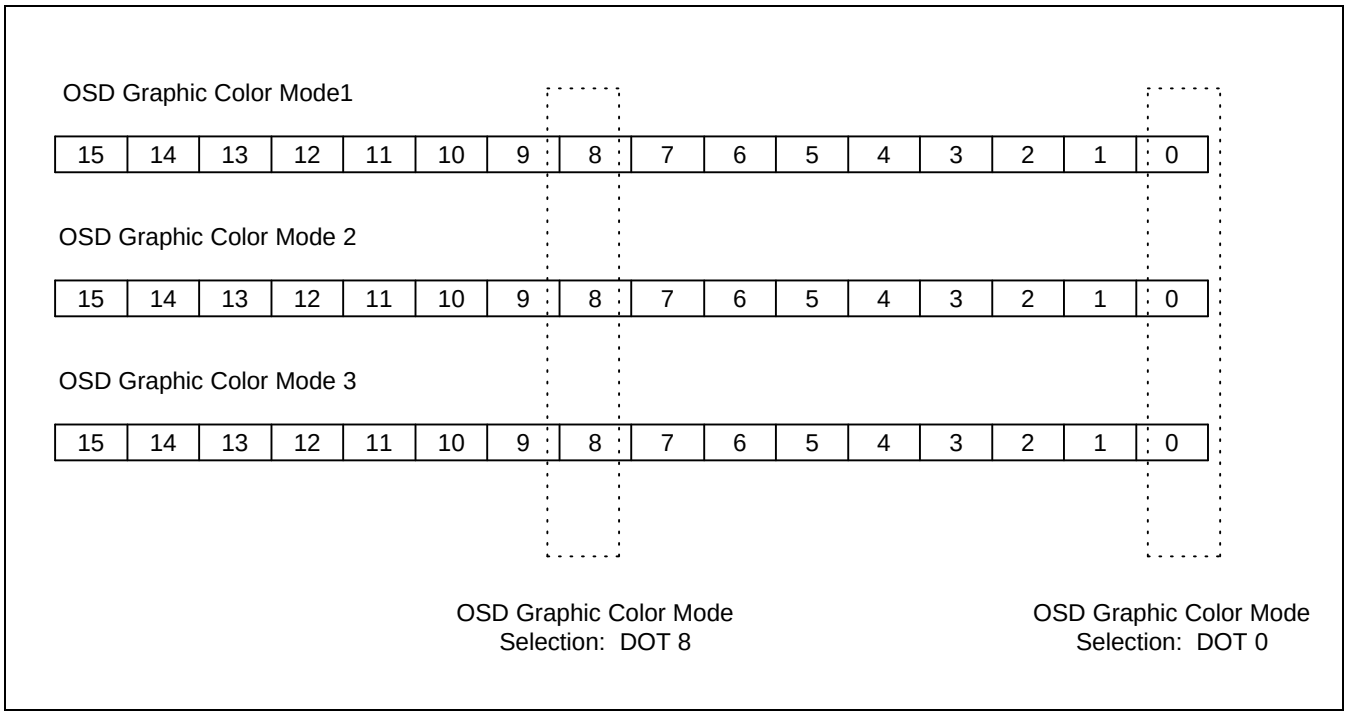


Figure 16-28. OSD Graphic Color Mode 1, 2, and 3

OSD COUNTER REGISTER (OSDCNT)

OSD counter register represents row counting (OSDCNT.8-11) and line counting (OSDCNT.0-5).

Row Counter Function

OSDCNT8-OSDCNT.11 to the OSD row read data. OSD row counter indicates the OSD row currently displayed. One row comprises one character (18 lines) and inter row space. The row counter value for the first row after a V-sync input is set to “0”. Low after a V-sync input is set to “0”.

Line Counter Function

Line counter shows in which line a characters is located. In OSD mode (OSDCON.0 = “0”), the counter counts 18 lines of characters and 15 line of inter-low space (OSDCMGN.0-3) and in CCD mode (OSDCON.0 = “1”, it counts 18 lines of characters and is lines of inter-row space (OSDCON.0-3). Both in OSD mode (OSDCON.0 = “0”) and CCD mode (OSDCON.0 = “1”), it counts 18 lines of characters and 15 lines of inter-row space (OSDVMGN.0-3)

Line counter lets you know in which line the currently display character is located. It is especially useful in line graphic OSD display.

Line counter can count upto 64 lines using (OSDCNT.0-5, cleared to “0” in line 0 of early row.

OSD COUNTER REGISTER

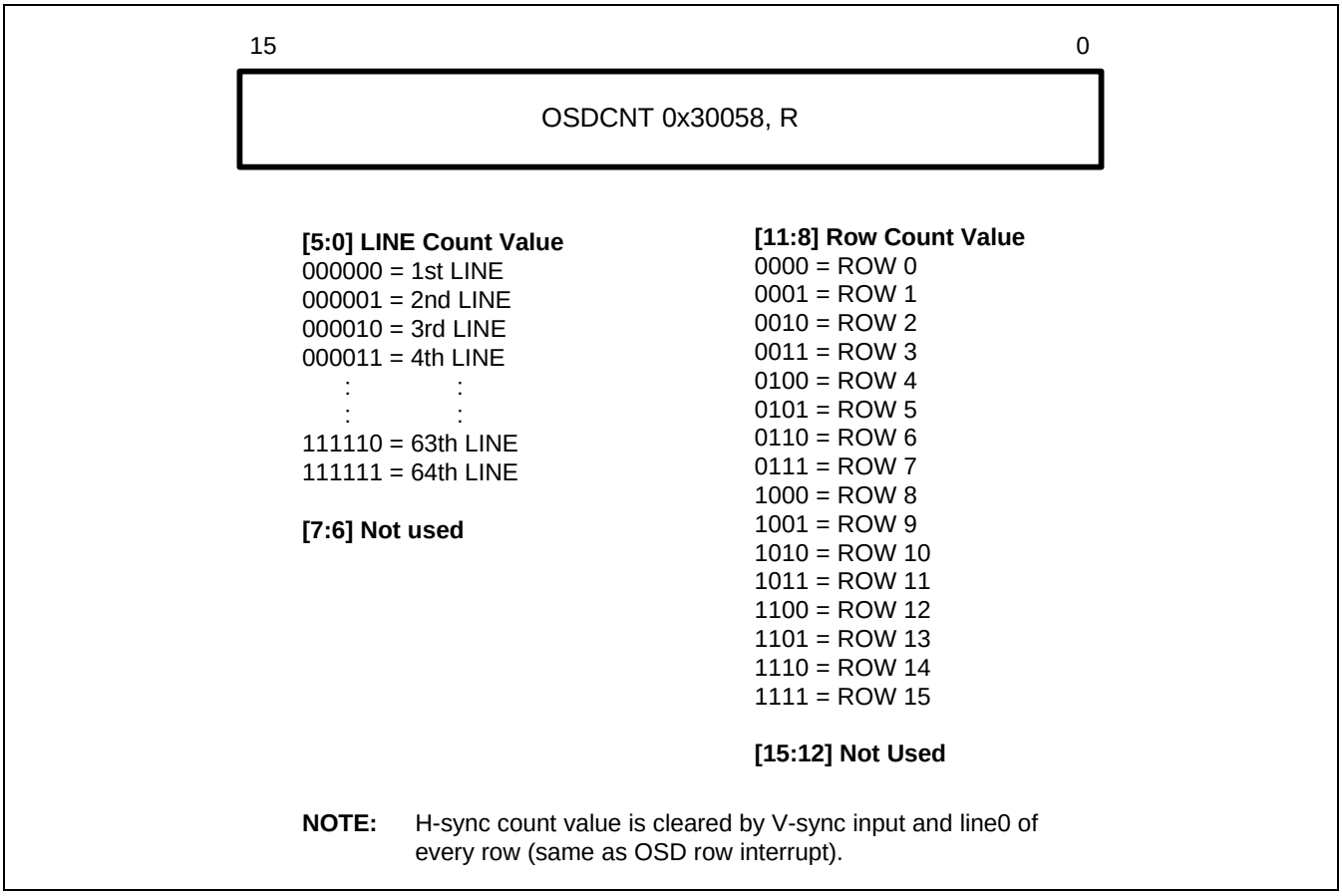


Figure 16-29. OSD Counter Register

OSD FIELD CONTROL REGISTER (OSDFLD)

OSD field control register helps recognizing whether the current field is an even field or on add one, in a TV signal frame. This control register must be defined for a correct field recognition of V-sync and H-sync entering the S3C380D. In order to recognize an even field, OSDFLD.0-3 defines the range starting from the point of V-sync edge, where H-sync must be present. If H-sync exits within the range, the field is recognized as an ever field.

OSDFLD.4 defines when H-sync must be detected. If it is set to “0”, the existence of H-sync is detected within the range set by OSDFLD.0-3 before V-sync is input. If it is set to “1”, it is detected after V-sync is input.

OSDFLD.5 describes whether the current field input by the field control which is set by OSDFLD.0-3 and OSDFLD.4 is an even field or an add one.

Field Detect by OSDFLD Control

When control register set is OSDFLD.0-3 = “1010B”, OSDFLD.4 = 0

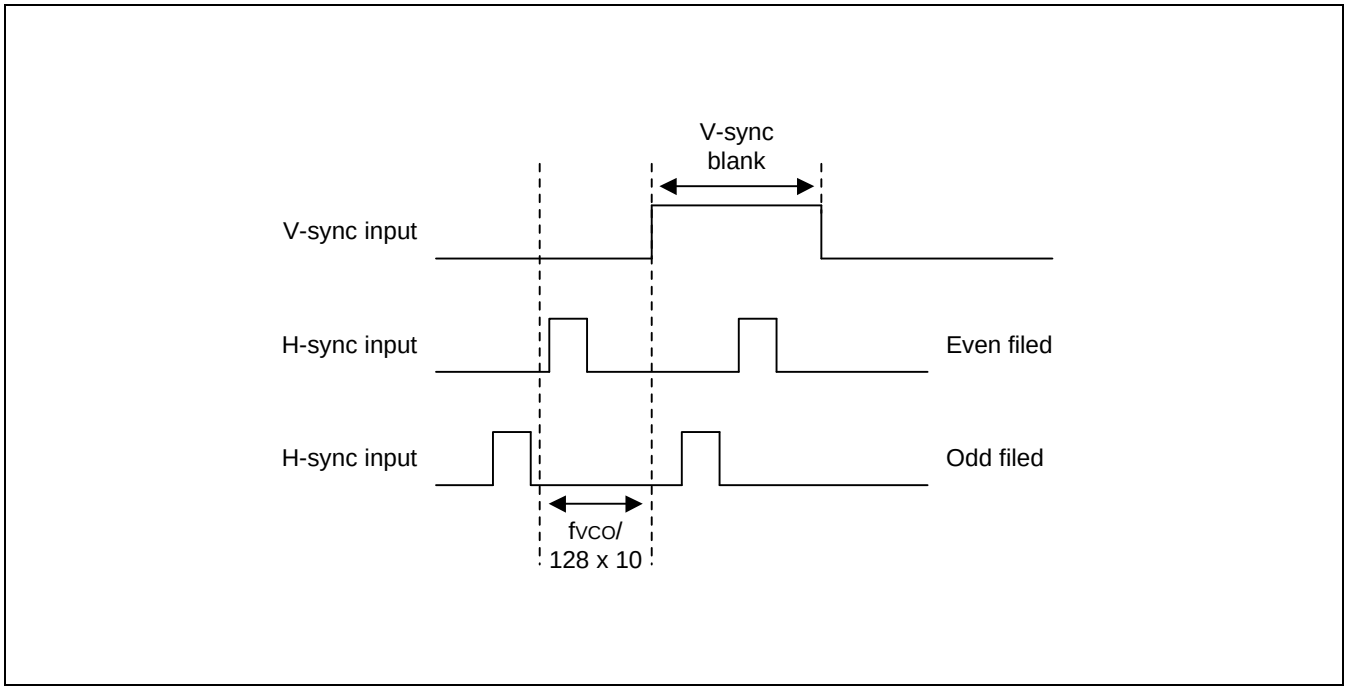


Figure 16-30. Field Detect in before V-sync

When control register set is OSDFLD.0-3 = “1010B”, OSDFLD.4 = 0

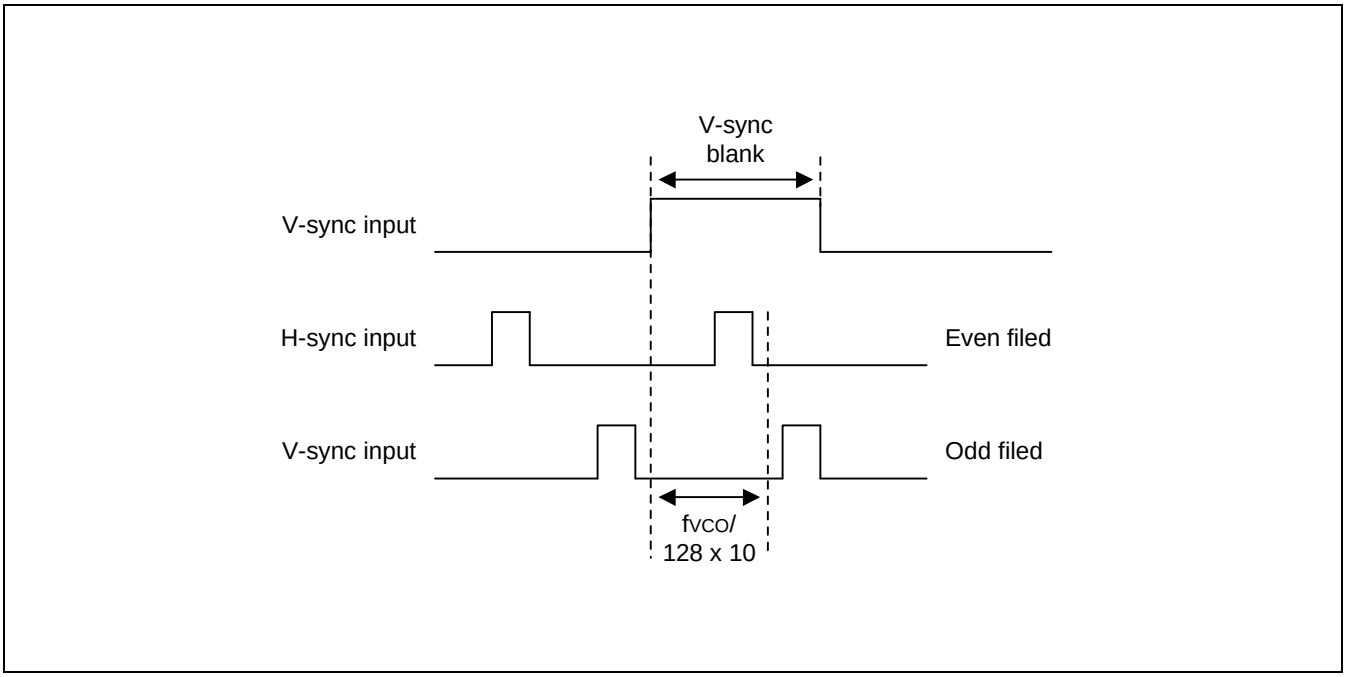
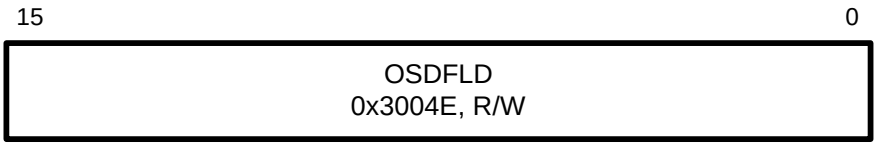


Figure 16-31. Field Detect in after V-sync



[3:0] Even Field Range
0000 = Not used
0001 = fvco/128 x 1
0010 = fvco/128 x 2
0011 = fvco/128 x 3
0100 = fvco/128 x 4
0101 = fvco/128 x 5
0110 = fvco/128 x 6 (Reset Value)
: :
: :
1101 = fvco/128 x 13
1110 = fvco/128 x 14
1111 = fvco/128 x 15

[4] H-sync Detect Position Select
0 = Detect H-sync Before V-sync
1 = Detect H-sync After V-sync

[5] Field Data (Read Only)
0 = Even Field
1 = Odd Field

[15:6] Not used

Figure 16-32. OSD Field Control Register

17 ELECTRICAL DATA

OVERVIEW

This chapter describes the S3C380D electrical data. Information is presented according to the following Table of Contents:

Table 17-1. Absolute Maximum Ratings

(T_A = 25°C)

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{DD}		– 0.3 to + 7.0	V
Input voltage	V_{I1}	P0.1-P0.3, P1.4-P1.7 (open-drain)	– 0.3 to + 6	V
	V_{I2}	All ports except V_{I1}	– 0.3 to V_{DD} + 0.3	
Output voltage	V_O	All output ports	– 0.3 to V_{DD} + 0.3	V
Output current high	I_{OH}	One I/O pin active	– 10	mA
		All I/O pins active	– 50	
Output current low	I_{OL}	One I/O pin active	+ 20	mA
		Total pin current for ports 0, 1, 2, and 3	+ 100	
Operating temperature	T_A	–	– 20 to + 85	°C
Storage temperature	T_{STG}	–	– 40 to + 125	°C

Table 17-2. D.C. Electrical Characteristics

(T_A = – 20°C to + 85°C, V_{DD} = 4.5 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Input high voltage	V_{IH1}	All input pins except V_{IH2}	0.8 V_{DD}	–	V_{DD}	V
	V_{IH2}	RESET	0.85 V_{DD}			
Input low voltage	V_{IL1}	All input pins except V_{IL2}	–	–	0.2 V_{DD}	V
	V_{IL2}	RESET			0.15 V_{DD}	
Output high voltage	V_{OH1}	V_{blank}, P2.4, P2.5 I_{OH} = – 1 mA	V_{DD}– 1.0	–	–	V
	V_{OH2}	All ports except V_{OH1} I_{OH} = – 500 uA	V_{DD}– 0.5			
Output low voltage	V_{OL1}	P2.4, P2.5 I_{OL} = 15 mA	–	–	1.0	V
	V_{OL2}	All ports except V_{OL1}, V_{OL3} I_{OL} = 2 mA			0.4	
	V_{OL3}	V_{blank} I_{OL} = 1 mA			0.4	
Input high leakage current	I_{LIH1}	V_{IN} = V_{DD} All input pins except I_{LIH2}	–	–	1	uA
	I_{LIH2}	V_{IN} = V_{DD} X_{IN}, X_{OUT}	3		20	
Input low leakage current	I_{LIL1}	V_{IN} = 0 V All input pins except I_{LIL2}	–	–	– 1	uA
	I_{LIL2}	V_{IN} = 0 V X_{IN}, X_{OUT}	– 3		–- 20	
Output high leakage current	I_{LOH1}	V_{OUT} = V_{DD} All output pins except I_{LOH2}	–	–	1	uA
	I_{LOH2}	V_{OUT} = 6 V P0.1-P0.3, P1.4-P1.7 (N-channel, open-drain)			10	
Output low leakage current	I_{LOL}	V_{OUT} = 0 V All output pins	–	–	– 1	uA

Table 17-2. D.C. Electrical Characteristics (Continued)

(T_A = - 40°C to + 85°C, V_{DD} = 4.5 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Pull-up resistor	R_{P2}	V_{IN} = 0 V RESET only	30	50	70	KΩ
Supply current	I_{DD1}	V_{DD} = 5 V 16 MHz CPU clock	–	50	100	mA
	I_{DD2}	Sleep mode		0.5	1	

Table 17-3. A.C. Electrical Characteristics

(T_A = - 40 °C to + 85 °C, V_{DD} = 4.5 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Interrupt input high, low width	t_{INTH}, t_{INTL}	Ports 2.0-2.3	–	300	–	ns
RESET input low width	t_{RSL}	Input	–	1000	–	ns
V-sync pulse width	t_{VW}	–	4	–	–	µs
H-sync pulse width	t_{HW}	–	3	–	–	µs
Noise filter	t_{NF1}	P2.0-P2.3	–	300	–	ns
	t_{NF4}	Glitch filter (oscillator block)		1000		
	t_{NF3}	RESET		1000		
	t_{NF2}	H-sync, V-sync		300		

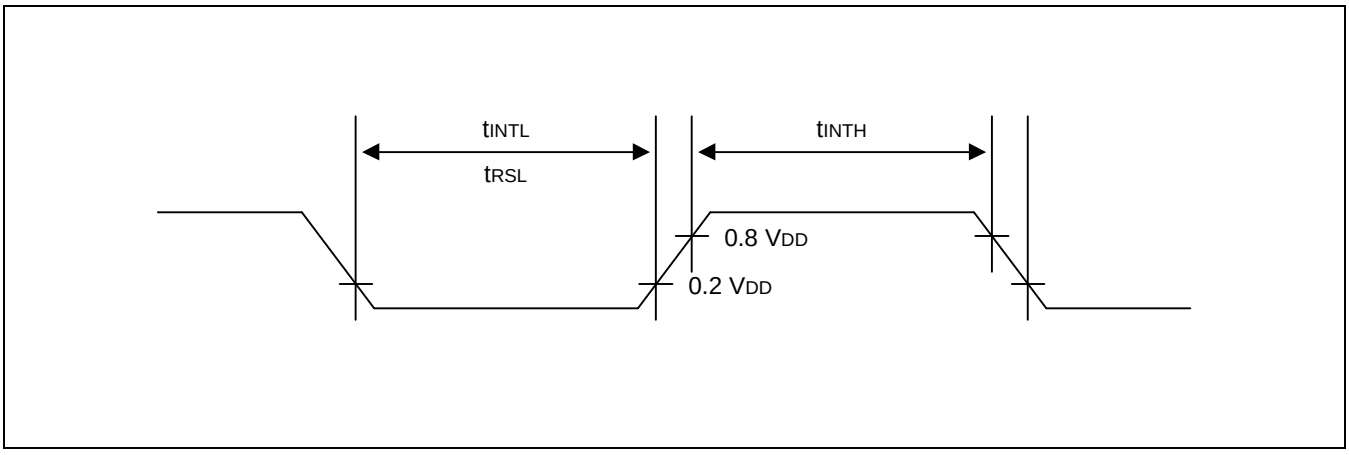


Figure 17-1. Input Timing measurement points

Table 17-4. Input/Output Capacitance

(T_A = – 40 °C to + 85 °C, V_{DD} = 0 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Input capacitance	C_{IN}	f = 1 MHz; unmeasured pins are returned to V_{SS}	–	–	10	pF
Output capacitance	C_{OUT}					
I/O capacitance	C_{IO}					

Table 17-5. Data Retention Supply Voltage in Sleep Mode

(T_A = – 20 °C to + 85 °C, V_{DD} = 4.5 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Data retention supply voltage	V_{DDDR}	Sleep mode	2	–	–	V
Data retention supply current	I_{DDDR}	Sleep mode V_{DDDR} = 5.0 V	–	–	2	mA

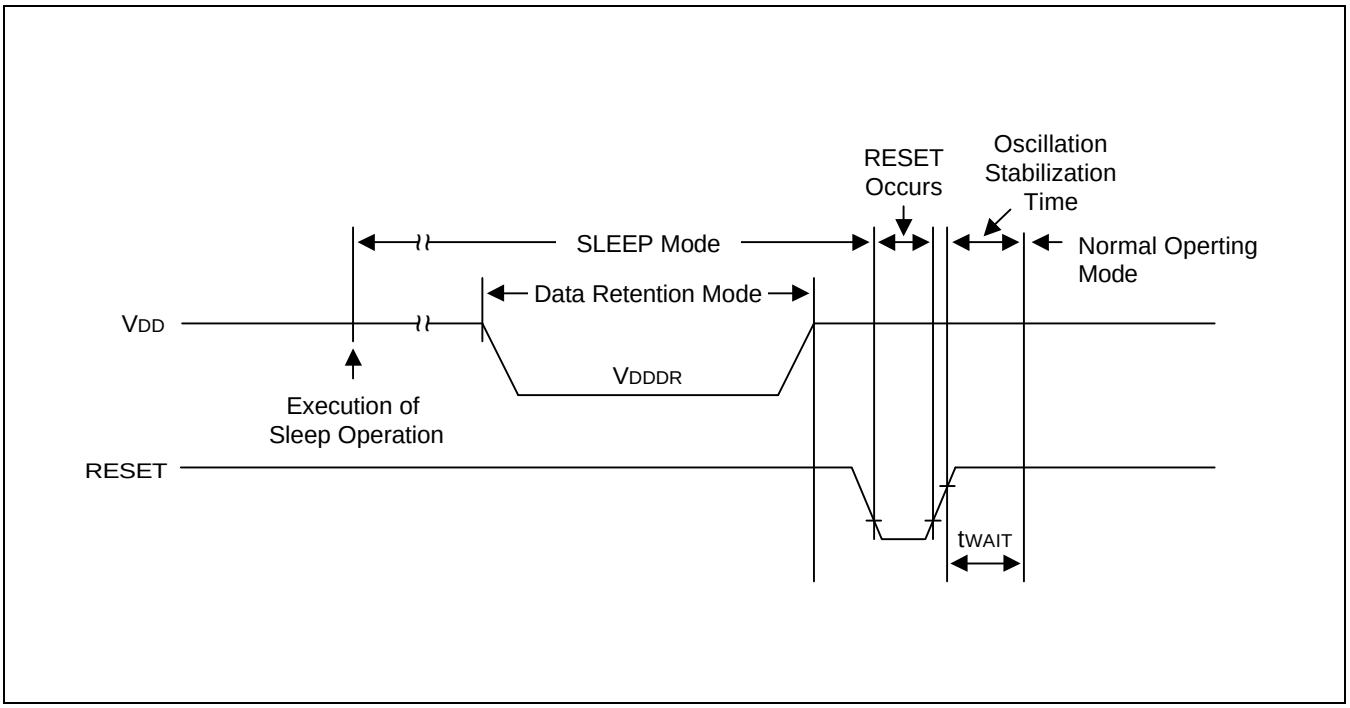


Figure 17-2. Sleep Mode Release Timing When Initiated by RESET

Table 17-6. Oscillator Frequency

(T_A = – 20 °C + 85 °C)

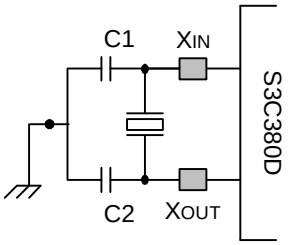
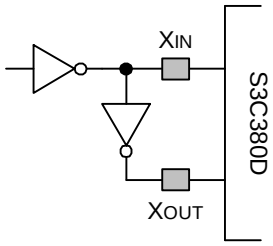
Oscillator	Clock Circuit	Test Condition	Min	Typ	Max	Unit
Crystal or ceramic		V_{DD} = 4.5 V to 5.5 V C1 = C2 = 33 pF recommended	–	32,768	–	Hz
External clock		V_{DD} = 4.5 V to 5.5 V	–	32,768	–	Hz

Table 17-7. Oscillator Clock Stabilization Time

(T_A = – 20 °C + 85 °C, V_{DD} = 4.5 V to 5.5 V)

Oscillator	Test Condition	Min	Typ	Max	Unit
Crystal	X_{IN} = 32,768 Hz	–	–	20	ms
External clock	X_{IN} input high and low level width (t_{XH}, t_{XL})	15	–	125	ns
Oscillator stabilization time	t_{WAIT} when released by a reset, X_{IN} = 32,768 Hz	–	–	500	ms
	t_{WAIT} when released by a interrupt (note)	–	–	4	ms

NOTE: The duration of the oscillator stabilization time, t_{WAIT}, when it is released by an interrupt, is determined by the settings in the basic timer control register, BTCON.

Table 17-8. A/D Converter Electrical Characteristics

(T_A = - 20 °C to + 85 °C, V_{DD} = 4.5 V to 5.5 V (ADC1-ADC4), V_{DD} = 5.0 V (ADC0))

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Resolution	–	–		–	–	4	Bit
Absolute accuracy (1)	–	CPU clock = 16 MHz	ADC0	–	–	± 1.0	LSB
			ADC1-4	–	–	± 0.5	LSB
Conversion Time (2)	t_{CON}	CPU clock = 16 MHz		–	(3)	–	ns
Analog input voltage	V_{IAN}	–	ADC1-4	AV_{SS}	–	AV_{REF}	V
			ADC0	1.5	–	2.0	V
Analog input impedance	R_{AN}	–		2	–	–	MΩ
Analog output impedance	R_{OAN}	CPU clock = 16 MHz Conversion time = 4 MHz		–	–	5	KΩ
		CPU clock = 16 MHz Conversion time = 0.5, 1, and 2 MHz		–	–	10	KΩ

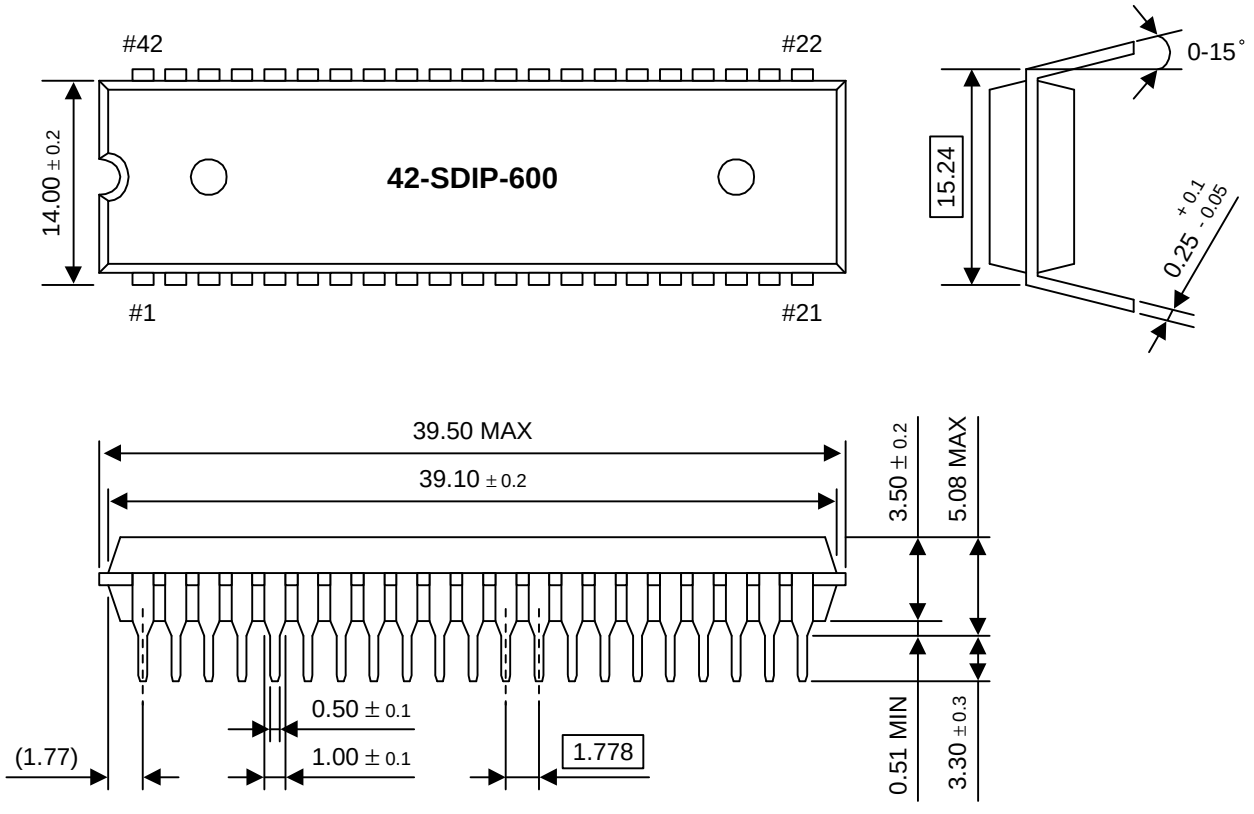
NOTES:

1. Excluding quantization error, absolute accuracy values are within ± 1 LSB (ADC0), ± 0.5 LSB (ADC1-4)
2. 'Conversion time' is the time required from the moment a conversion operation starts until it ends
3. ADC conversion time is controled by ADCON.9-.8.

18 MECHANICAL DATA

OVERVIEW

The S3C380D microcontroller is currently available in 42-pin SDIP (42-SDIP-600) package.



NOTE: Dimensions are in millimeters.

Figure 18-1. 42-Pin SDIP Package Dimensions

19 S3F380D MTP

OVERVIEW

The S3F380D single-chip CMOS microcontroller is the MTP (Multiple Time Programmable) version of the S3C380D microcontroller. It has an on-chip Flash ROM instead of a masked ROM. The flash ROM is accessed by serial data format.

The S3F380D is fully compatible with the S3C380D, both in function and pin configuration.

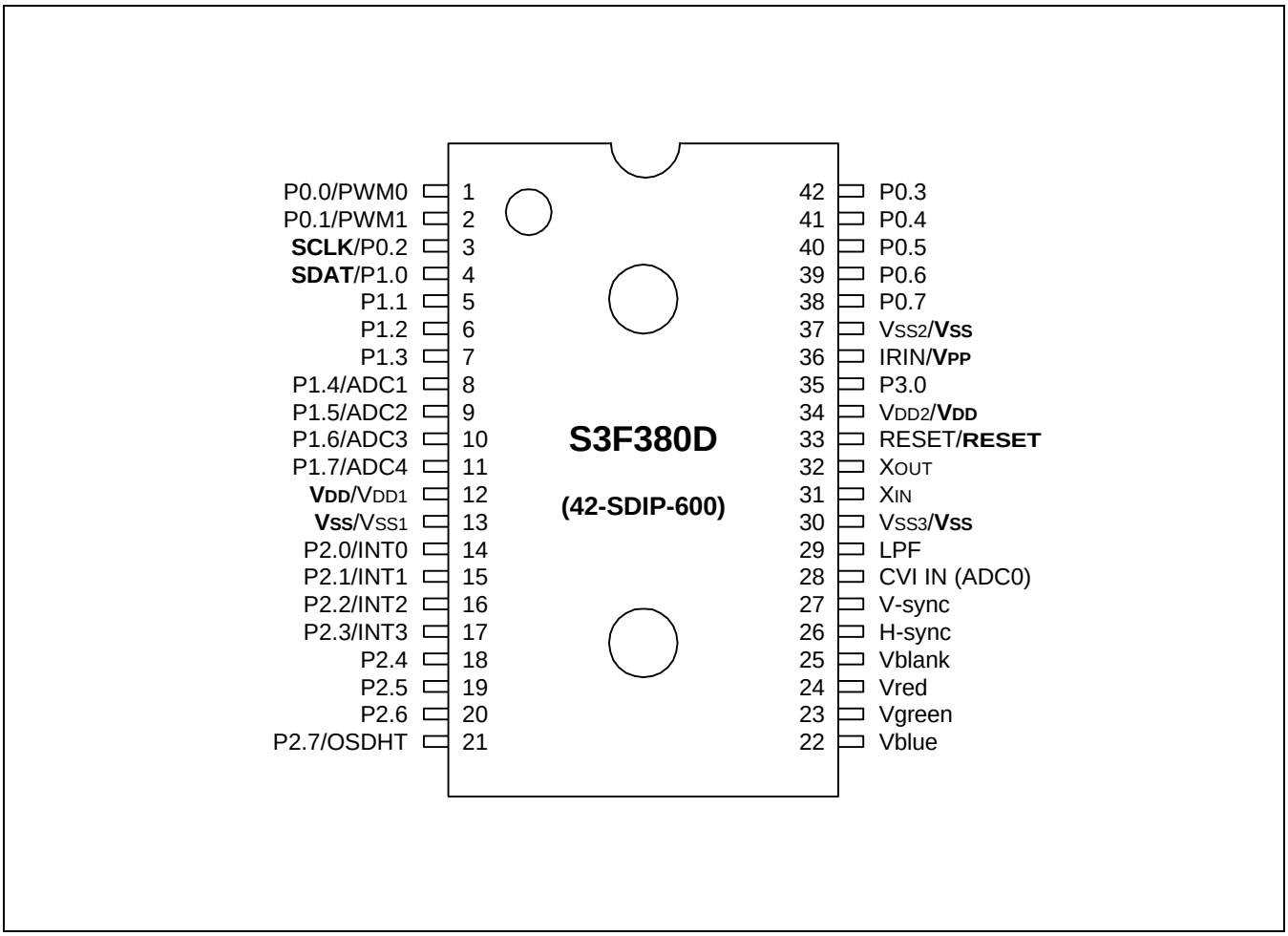


Figure 19-1. S3F380D Pin Assignment (42-SDIP)

Table 19-1. Descriptions of Pins Used to Read/Write the Flash ROM (S3F380D)

Main Chip	During Programming			
Pin Name	Pin Name	Pin No.	I/O	Function
P1.0 (Pin 4)	SDAT	4	I/O	Serial data pin (output when reading, Input when writing) Input and push-pull output port can be assigned
P0.2 (Pin 3)	SCLK	3	I/O	Serial clock pin (Input only pin)
IRIN	V_{PP}	36	I	0-5 V: operating mode 12.5 V: MTP mode
RESET	RESET	33	I	5 V: operating mode, 0 V: MTP mode
V_{DD}/V_{SS}	V_{DD}/V_{SS}	12/34, 13/30/37	I	Logic power supply pin.

Table 19-2. Comparison of S3F380D and S3C380D Features

Characteristic	S3F380D	S3C380D
Program Memory	128-Kbyte Flash ROM	128-Kbyte mask ROM
Operating Voltage (V_{DD})	4.5 V to 5.5 V	4.5 V to 5.5 V
MTP Programming Mode	V_{DD} = 5 V, V_{PP} = 12.5 V	–
Pin Configuration	42 SDIP	42 SDIP
Flash ROM programmability	User program under 100 time	Programmed at the factory