GP2000

GPS Chipset – Designer's Guide

The GP2000 chipset comprises the GP2015 RF front end and the GP2021 12-channel correlator. Together with a microprocessor and a SAW filter (both available from Mitel Semiconductor) these two circuits form the heart of a GPS (Global Positioning System) Receiver. This designer's guide provides technical information on the GP2000 chipset, on GPS systems in general and on the GPS Architect™ development kit.

PRACTICAL RECEIVER IMPLEMENTATION

A typical GPS receiver is shown in Fig. 1. From the antenna, the signal is down-converted by the front end circuit ready for processing by the correlator. A microprocessor controls the receiver operation, including interfacing to any external display or other function.

SUBSYSTEM SELECTION ISSUES

Microprocessor Selection

It is difficult to state a minimum specification for a microprocessor suitable for use in a GPS receiver since the choice is influenced by many factors.

The performance requirements are dictated by the application and functionality of the receiver as a whole and possibly by the system of which the receiver is a part.

For volume GPS applications the trend is towards higher levels of integration. As a consequence there is a need to share resources with other parts of the complete system.

The microprocessor may be required to perform processing of tasks which are not directly related to producing a position fix, such as control of devices peripheral to the receiver.

Spare processing power, after catering for the GPS specific tasks, is an attractive feature for providing cost reductions by the possible removal of other microprocessors.

Normally, the microprocessor is required to perform both integer and floating point arithmetic, although floating point calculations will usually be constrained to the navigation solution procedures.

The use of a maths co-processor is usually precluded for cost reasons but may be unavoidable if, for instance, a high solution update rate is required (many times a second).

Processor loading can be traded-off against the number of active channels. However, retrospectively, this may have an adverse impact on the software complexity and navigation performance.

The familiarity of the developers with the development platform and the suitability of the available tools for the development of real-time embedded applications is also important.

Memory Selection

The memory requirements of the GPS receiver should be carefully considered since this constitutes a large portion of the component costs. The choice of memory can be complex and trade-offs need to be made against component costs, speeds and power consumption. As with the choice of microprocessor, the size of memory required by the receiver is also application specific.

However, typical memory sizes, when considering just the GPS software, are 256 Kbytes of ROM and 64 Kbytes of RAM. (These figures are for a production engine: a development system like GPS Architect uses more RAM – 512KBytes).

Memory savings by the reduction of the number of active channels are not necessarily significant.

Some non-volatile memory (NVM) may be required for information retention during periods of power-down. Parameters such as time-to-first-fix (TTFF) can be substantially improved by retaining satellite almanacs, ephemerides, reference oscillator characteristics and other data in the NVM. Typical NVM requirements amount to about 8 Kbytes.

The choice of RAM type may be influenced by software throughput requirements.

Time critical portions of code may be run from fast SRAM (zero wait state) at the expense of memory cost and power consumption.
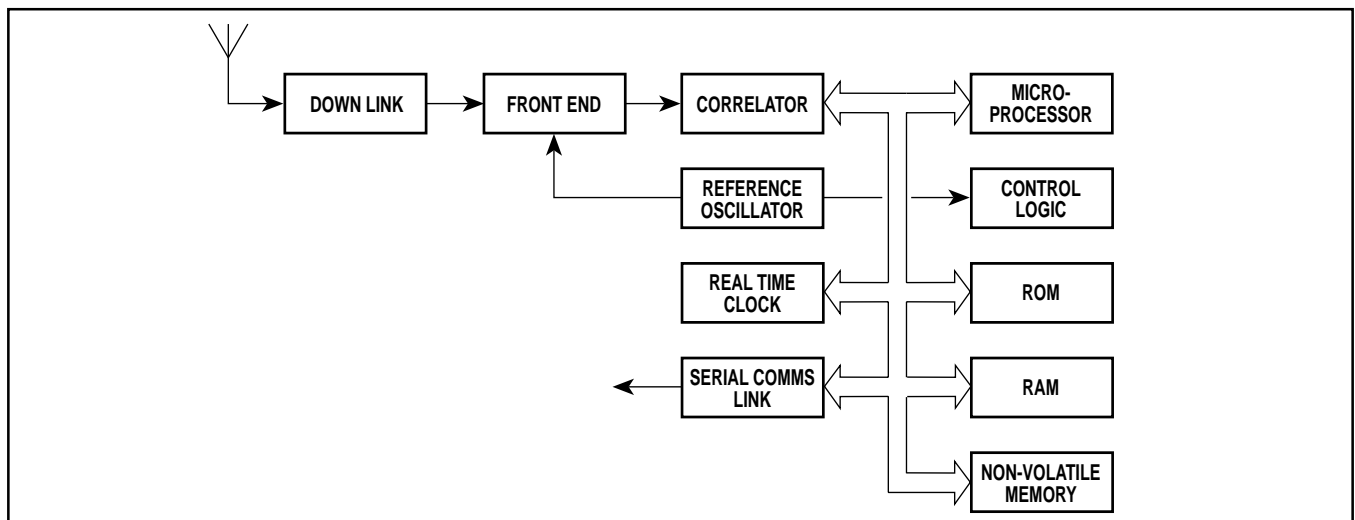

Fig. 1 Generic GPS receiver

The choice of microprocessor may influence memory selection if it has desirable features such as an on-chip cache RAM.

Reference Oscillator

The reference oscillator can be a significant cost component in a GPS receiver.

The short-term stability tends to impact on the receiver tracking performance since it is equivalent to the receiver undergoing a change in dynamics.

The long-term stability tends to impact signal acquisition times and receiver TTFF since it can widen the acquisition search window and increase the window overlap.

The main trade-offs for reference oscillator selection are between cost, stability and temperature range.

Careful consideration should be given to the operating temperature range due to its relationship to oscillator cost (a wider temperature range for a given stability means a higher cost) and stability (a narrower temperature range for a given stability at lower cost) .

Temperature compensated crystal oscillators (TCXOs) of stabilities in the region ±3ppm over a temperature range of approximately −30°C to +70°C provide a good general purpose oscillator. They can give acceptable performance even without software characterisation of the oscillator (prediction of the oscillator's offset from its nominal value).

Cheaper, less stable (±10ppm or so) uncompensated crystal oscillators can be used in conjunction with software characterisation and compensation and still give the required performance over a wide operating temperature range.

The designer should be aware if the type of oscillator used is prone to any sudden step changes in stability over the operating range since such characteristics are likely to cause loss of signal tracking.

Other Blocks

Real-Time Clock

The use of a real-time clock, if kept active during power-downs, can shorten acquistion times by enabling prediction of satellite visibilities and Doppler shifts. This reduces the TTFF by enabling a 'warm' start.

Serial Communications

Most GPS engines will have at least two serial I/O ports. One is usually dedicated to the reception of RTCM SC-104 DGPS correction data and the other for receiver control and status monitoring.

Control Logic

Dependent upon the exact nature of the receiver components, a certain amount of control or glue logic may be necessary to interface the components together.

GP2015/GP2021/ARM60 GPS RECEIVER

Fig. 2 shows how the GP2015, GP2021 and ARM60 can be used to create a low cost GPS receiver solution with 12 channels and a minimum of external supporting components.

The internal memory management control logic of the GP2021 means that when used with the ARM60 no additional control or glue logic is required.

The GP2021 supports RAM, E²PROM and ROM/EPROM/FLASH with a configurable number of wait states for the E²PROM and ROM. The GP2021 can only support zero wait state RAM.

The on-chip real-time clock (RTC) and dual UART functions reduce the need for external components still further.

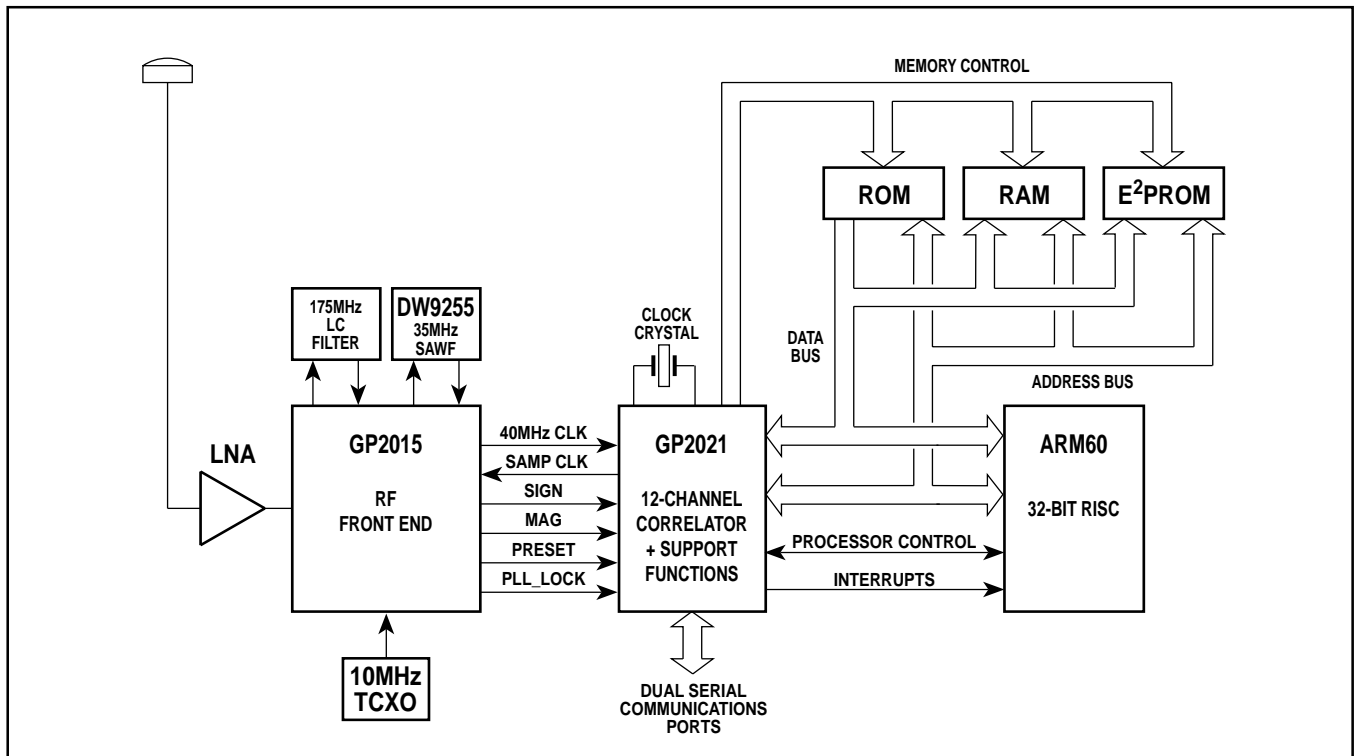The GP2015 and GP2021 integrated circuits are described in the following two sections

Fig. 2 GP2015/GP2021/ARM60 GPS receiver

GP2015 RF FRONT END

FEATURES

- ■ L1 C/A Code Front End
- ■ Low Voltage Operation (3V to 5V)
- ■ Low Power Consumption (200mW at 3V)
- ■ On-chip Phase Locked Loop including VCO
- ■ 3-Stage Down Conversion

- ■ 2-Bit Digital Output (Sign and Magnitude)
- ■ −40°C to +85°C Operating Temperature Range
- ■ Interfaces to GP2021 Digital Correlator
- ■ Few External Components Required
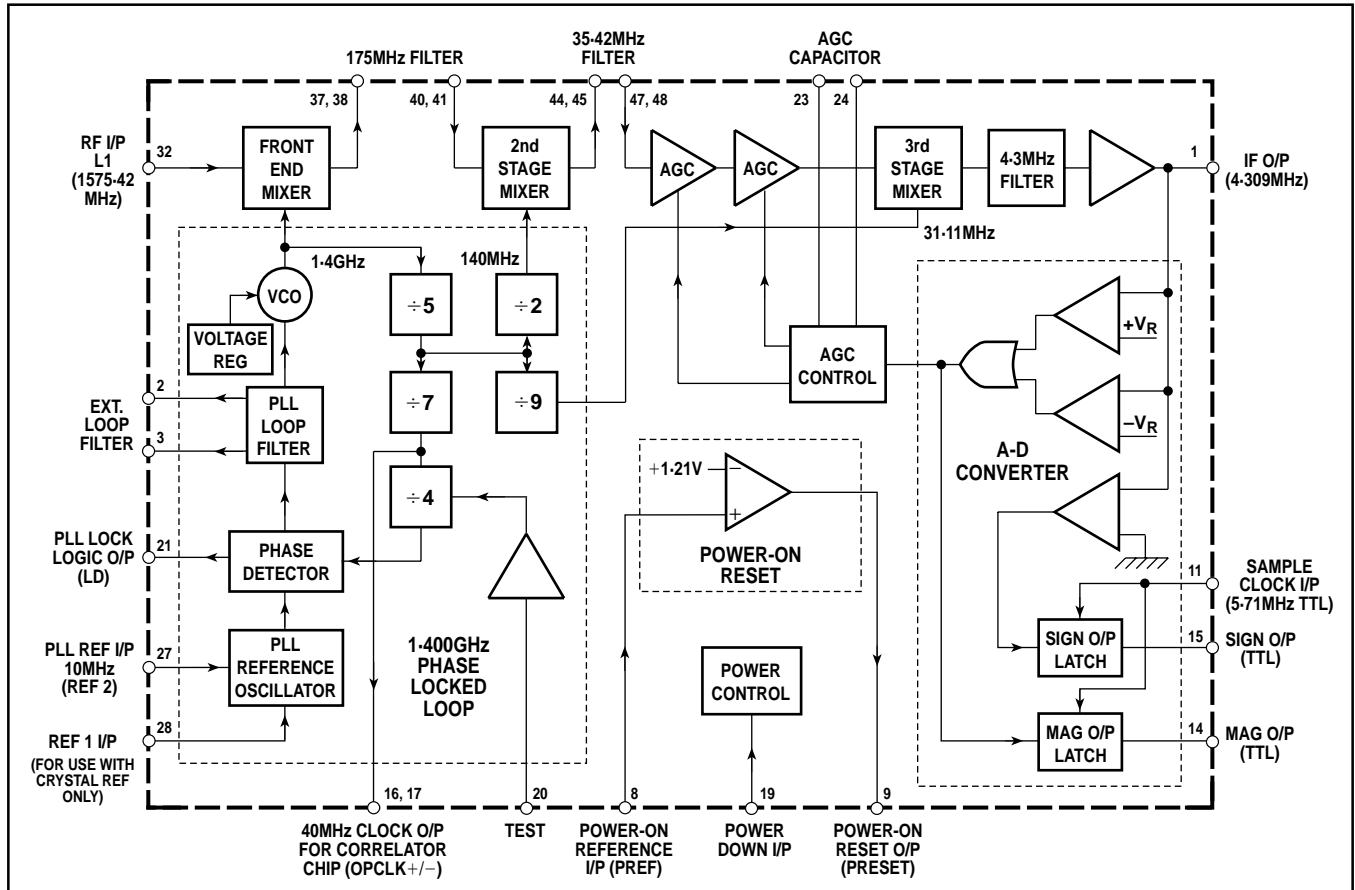
FUNCTIONAL OVERVIEW

Fig. 3 GP2015 block diagram

Fig. 3 is a block diagram of the front end integrated circuit GP2015.

Several key blocks can be identified:

- On-Chip Phase Locked Loop
- Triple Down-Conversion
- AGC
- Power Control

On-Chip Phase locked Loop

An on-chip Phase Locked Loop (PLL) requiring only an external 10MHz reference (0·1 to 1·2V p-p) synthesises 1400MHz which is used to generate all intermediate frequency local oscillators for signal down-conversion. These frequencies are 1400MHz, 140MHz and 31·111MHz. An external 40MHz clock (for use by the GP2021 correlator) is also generated. This signal is low level differential to minimise interference.

The design has been implemented to minimise the number of external components. The application circuit (Fig. 4) shows the components required.

The Voltage Controlled Oscillator (VCO) has an on-chip voltage regulator to improve noise immunity of the PLL (only available in 5V operation).

Signal Down-Conversion
First Stage

The first stage of signal down-conversion mixes the L1 signal at 1575·42MHz with a local oscillator at 1400·0MHz to give a first IF of 175·42MHz. This places the image at 1224·58MHz (i.e. approximately L2)

The high first IF means that image attenuation can be easily achieved with a combination of a selective antenna and a simple low-cost filter at the antenna or input to the front end.

The first stage mixer has an image rejection filter of about 5dB (minimum) and 1dB compression point of −22dBm (minimum). Together with the selective antenna and RF filter, this provides a good level of protection against saturation at the input from out-of-band interfering signals.

The first IF filter is also used to suppress interfering signals near to the IF to prevent saturation of the second stage mixer.

In practice, a typical first IF filter can be realised with discrete components as, for example, a 2-pole coupled-tuned filter, where a bandwidth of about 20MHz is achievable.
Second Stage

The second stage of signal down-conversion mixes the signal at 175·42 MHz with a local oscillator at 140·0 MHz to give a
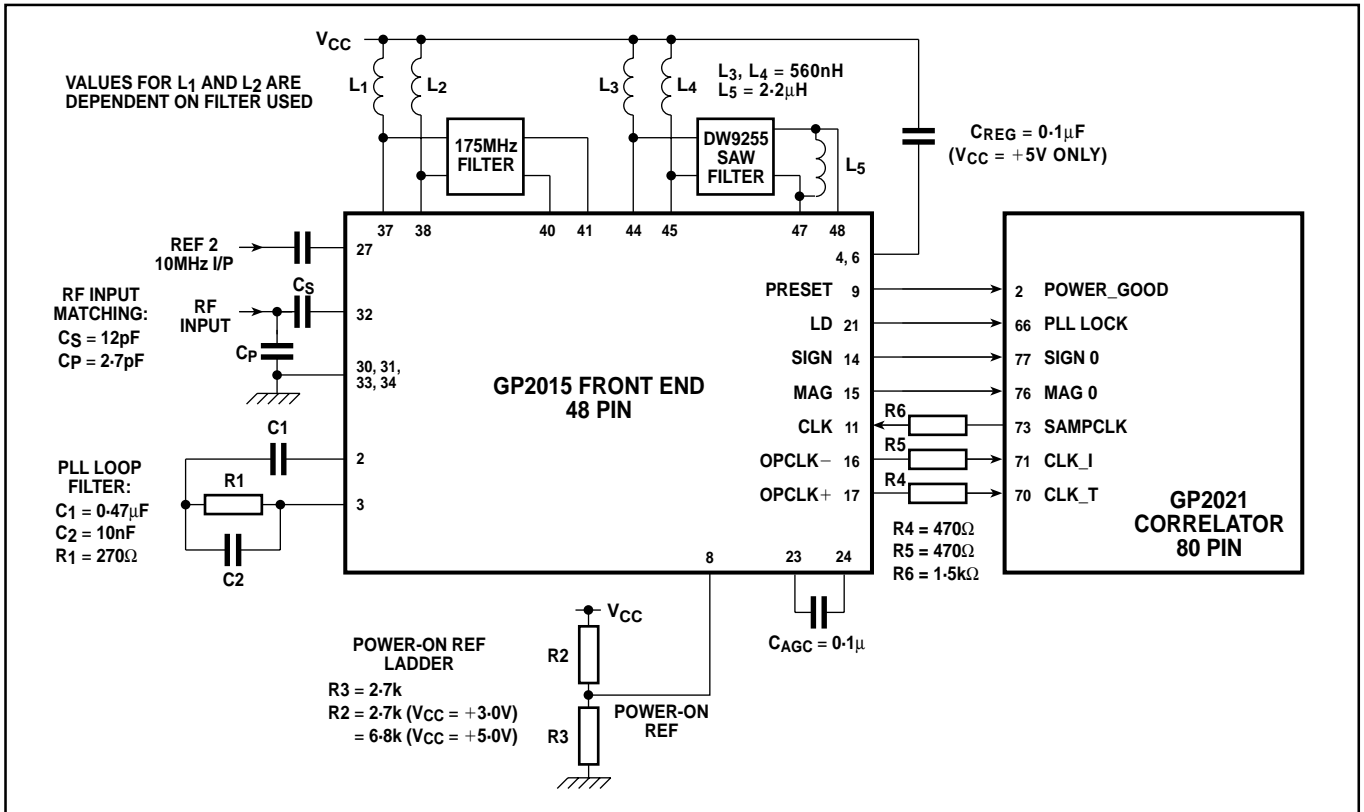
3

Fig. 4 Application circuit

second IF of 35·42 MHz. This places the image at 104·58MHz.

Attenuation of the image is achieved via a 175·42MHz bandpass (±1MHz) filter before the second stage mixer. This complements any image attenuation at this frequency from the antenna and RF filter.

Third Stage

The third stage of signal down-conversion mixes the signal at 35·42MHz with a local oscillator at 31·111MHz to give a third IF of 4·309MHz, placing the image at 26·802MHz.

The second IF filter is critical to system performance and effectively sets the bandwidth of the front end. To achieve high levels of image and interference rejection, a filter with a very tight passband and sharp cut-off is required.

The use of a SAW filter, such as the GEC Plessey Semiconductors DW9255, is recommended to achieve these types of characteristic. The DW9255 has a 1·9MHz passband centred on 35·42MHz with 0·8dB of passband ripple and out-of-band rejection of better than 21dB at ±2·0MHz and better than 35dB at ±7·5MHz.

AGC and Sampling

Following the third stage mixer the signal is filtered with an on-chip 4·3MHz passband (±1MHz) filter. The signal is then used to control the gain of the AGC which appears between the second IF filter and third stage mixer (Fig. 5). The AGC caters for factors such as varying RF amplifier gain and cable loss.

The AGC operates such that the magnitude bit (MAG) at the output of the 2-bit A-to-D converter is high for nominally 30% of the time. When the magnitude bit is high it is given a value of 3. When low it is given a value of 1. The SIGN bit will be high for nominally 50% of the time. This statistical distribution of data aids the system in suppressing CW interference.

Signal Down-Conversion Sampling

The signal at its third IF of 4·309MHz is then sampled and latched at the SIGN and MAG outputs. When used with the GP2021 the sampling clock is at 5·714MHz. The aliasing present in the sampling process produces a final digital signal at an IF of nominally 1·405 MHz (Fig. 6).

Power Supplies

The analog and digital sections of the device can be operated from separate power supplies to prevent interaction with digital switching transitions in the analog section.

The GP2015 includes an on-chip voltage detector for the digital supply. This drives a logic output which goes high when the power supply has reached a nominal value. This output can be used to disable the correlator and microprocessor when power supply switching is occurring.

The GP2015 can be put into a Power-Down mode where most of the device, except the power supply switching detector, is disabled. This changes the current consumption from nominally 70mA to 10mA.
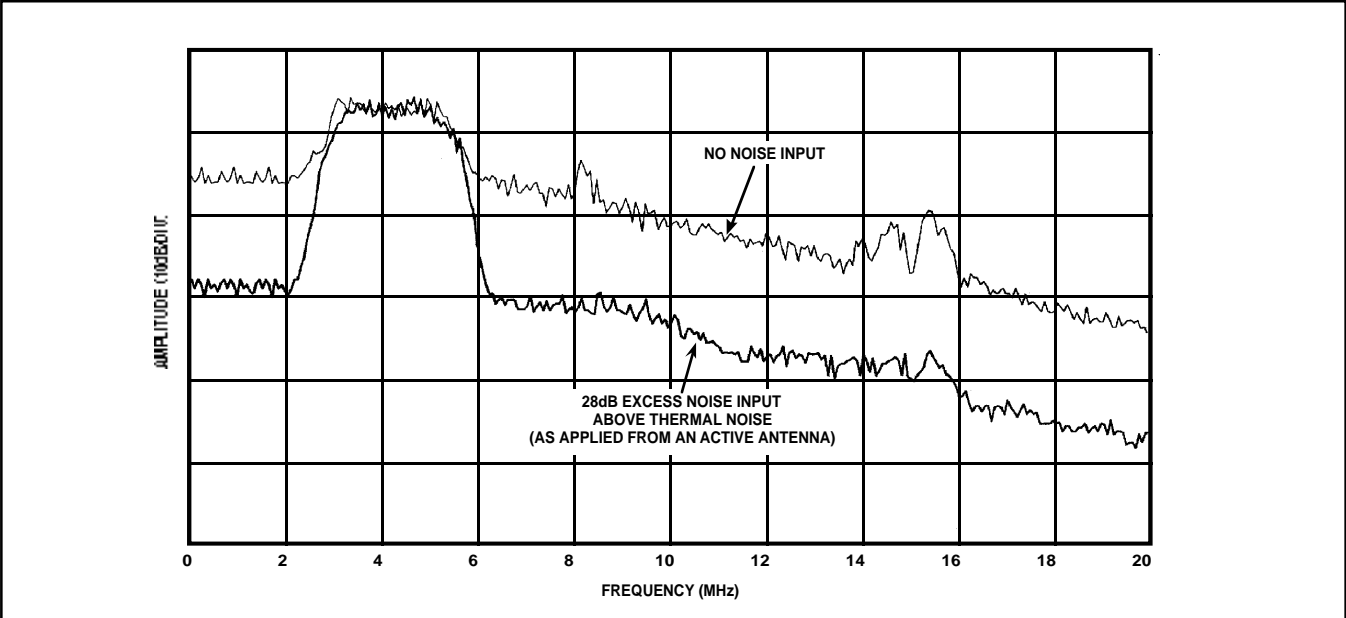
Fig. 5 GP2015 IF OUT spectra, sowing effect of AGC in the presence of applied input noise (GP2015 on GPS Orion Receiver Board)
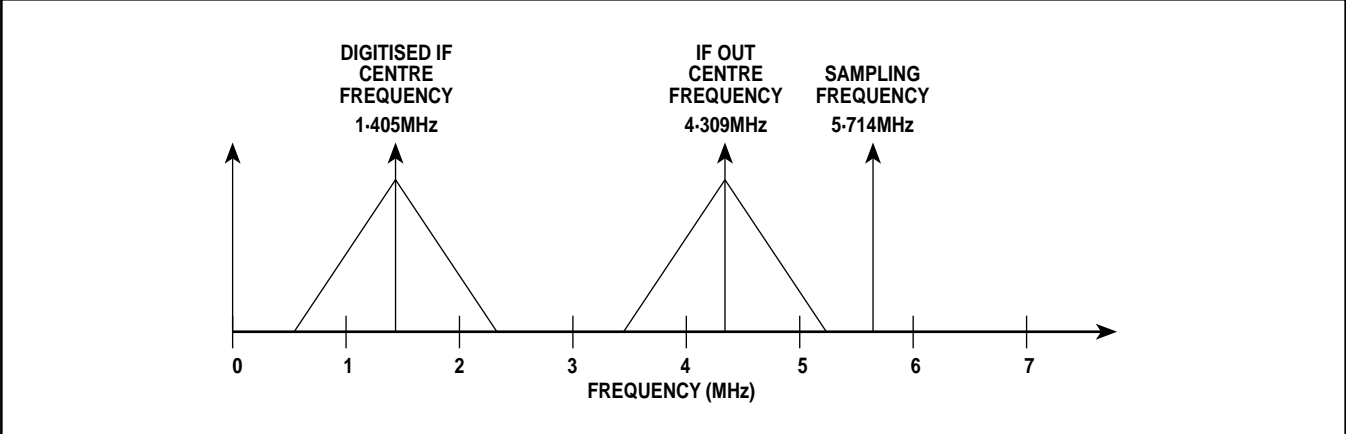

Fig. 6 GP2015 signal down-conversion sampling

GP2021 DIGITAL CORRELATOR ARCHITECTURE

FEATURES

- 12 Fully Independent Correlation Channels
- On-chip Dual UART and Real-time Clock
- Compatible with 16-bit and 32-bit Microprocessors
- Memory Control Logic for the ARM60 Microprocessor

- Compatible with GP2010 and GP2015 RF Front Ends
- Power Dissipation 150mW Typical
- Low Voltage/Current Power-down Mode
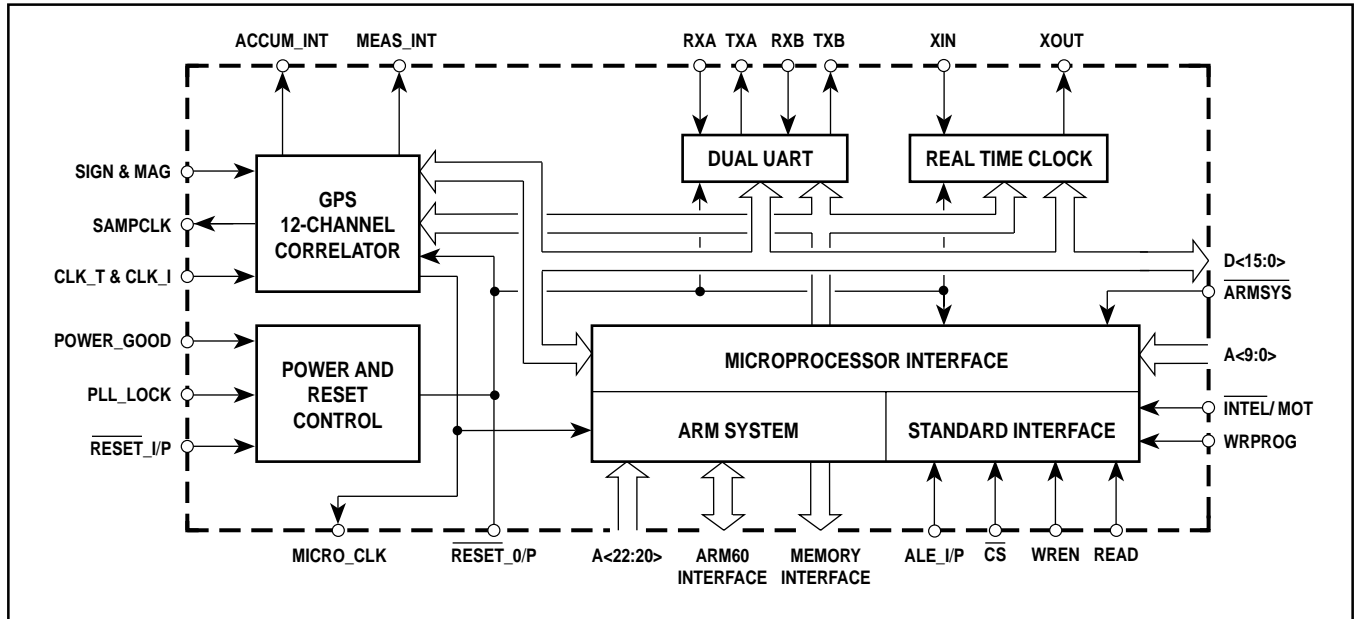- Battery Back-up Voltage 2·2V (Minimum)

FUNCTIONAL OVERVIEW

Fig. 7 GP2021 block diagram

The GP2021 (Fig. 7) can have all channels configured independently. Each channel can also be disabled for power saving and microprocessor bandwidth purposes.

The GP2021 accepts 1-bit or 2-bit digital signals through one of two pairs of selectable inputs so that it can be used with up to two antennas. It can also be configured to accept input signals in real (both components in-phase) or complex (one component in-phase, the other component quadrature phase) form.

The GP2021 supplies two programmable interrupts for accessing the accumulation and measurement data.

The 40MHz clock required by the correlator chip can be derived from a GP2010 or GP2015 front end.

Microprocessor Interface

The GP2021 is compatible with 16- and 32-bit microprocessors. It supports four styles of microprocessor:

- ARM60
- Motorola
- Intel 80186
- Intel 486

In ARM60 mode, on-chip logic provides direct interfacing to the ARM60 microprocessor and the system's memory without any additional supporting hardware. For other microprocessors, additional external logic may be required.

Dual UART

There are two independent UARTs on the GP2021 which have programmable data rates, for both transmit and receive,

between 300 and 76·8Kbaud. The parity bit can be programmed to be odd, even or none.

There are 8 byte deep FIFOs on both transmit and receive. The UARTs are accessed by polling

Real Time Clock

The real time clock (RTC) uses a 32kHz watch crystal and can be used to maintain a measure of elapsed time during GP2021 Power-down mode. This can be used to improve the TTFF by enabling prediction of satellite visibility and Doppler frequency offsets from the estimated receiver location.

Using a 24-bit 1-second counter, the RTC has a range of approximately 194 days.

The RTC Block also contains a Watchdog which causes a chip reset if it is not accessed for approximately 2 seconds. The Watchdog function can be disabled.

Power and Reset Control

The GP2021 has a Power-down mode which allows the supply voltage to drop to 2·2V (minimum). In this mode all GP2021 functions are disabled except for the RTC. During power-down the microprocessor clock is maintained to the end of the current clock cycle (falling edge) to prevent corruption of battery backed RAM, and all inputs (except those relevant to the Power and Reset Control block) are clamped to known logic levels to prevent power consumption by extraneous switching. There is no requirement for external pull-up or pull-down resistors. A GP2021 reset can also be initiated by use of the front end PLL lock indicator pin.
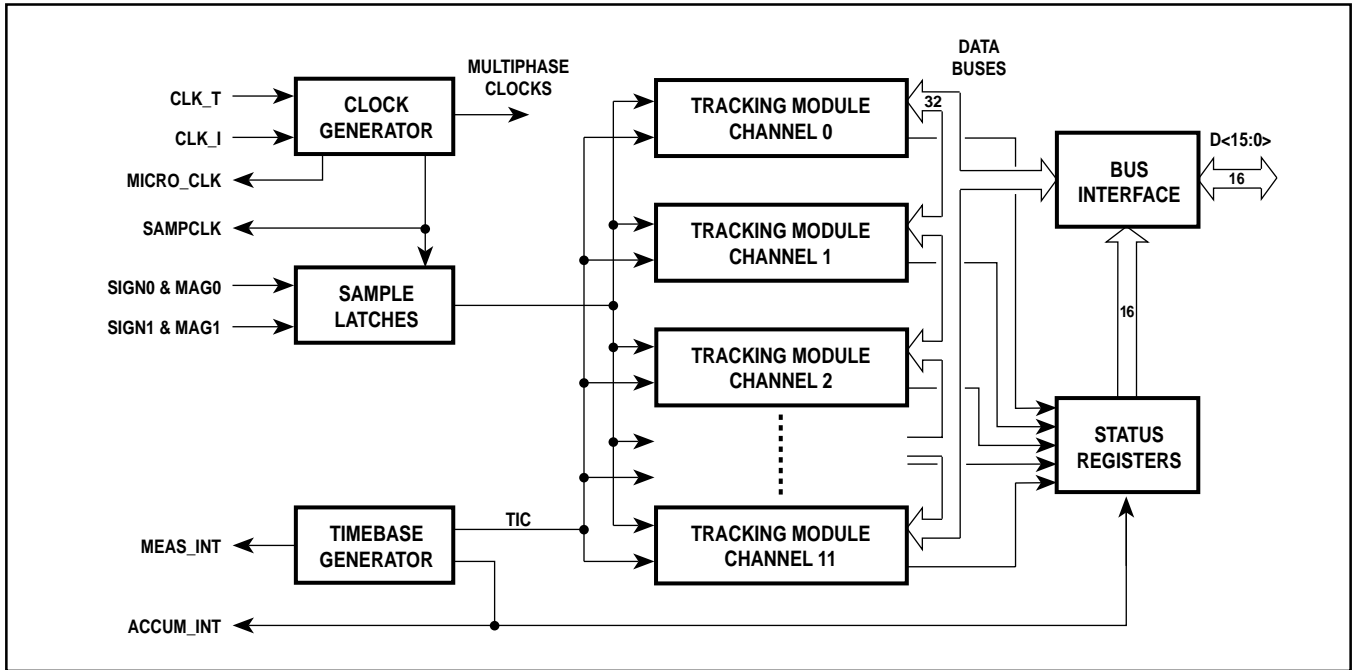
Fig. 8 GP2021 correlator block diagram

CORRELATOR DESCRIPTION

The GP2021 Correlator block is shown in Fig. 8; the constituent elements are described in the following sections.

Clock Generator

The Clock Generator accepts a differential input clock, CLK_T/CLK_I, normally at 40MHz. From this signal it generates:
(i) Multi-phase clocks for internal use by the rest of the device.
(ii) MICRO_CLK (pin 31): a 20MHz clock with a 1:1 mark-space ratio provided as an output for a microprocessor which has its own memory manager/state machine.
(iii) MCLK (pin 30): a clock for the ARM60 microprocessor, which incorporates stretched phases to implement wait-states for the wait-state memory. This is generated by the internal Microprocessor Interface, which is especially configured for the ARM60 when NARMSYS (pin 4) is set low.
(iv) SAMPCLK: a 5·714MHz clock for internal sampling of the input signal and also provided as an output for possible use by the front end as a sample clock.

Timebase Generator

The Timebase Generator generates:
(i) ACCUM_INT: a programmable interrupt which is normally used as a trigger to access new accumulation data from the tracking modules. The default interrupt period is 505·05µs for a master clock of 40MHz.
(ii) TIC: a signal of programmable period which is used to sample and latch the Tracking Modules measurement data – all at the same instant. The default TIC period is 0·0999999 seconds for a master clock of 40MHz.
(iii) MEAS_INT: an interrupt generated from TIC which is normally used to access new measurement data from the measurement data registers or as a timebase for switching software module tasks.

Sample Latches

The input signals at SIGN0, MAG0, SIGN1 and MAG1 are sampled (normally at 5·714 MHz) and latched for distribution to the Tracking Modules.

Bus Interface

The Bus Interface controls data transfer between the external 16-bit and the internal 32-bit data bus.

Status Registers

This block consists of four registers, three registers containing status information relevant to the accumulation data and one register relevant to the measurement data.
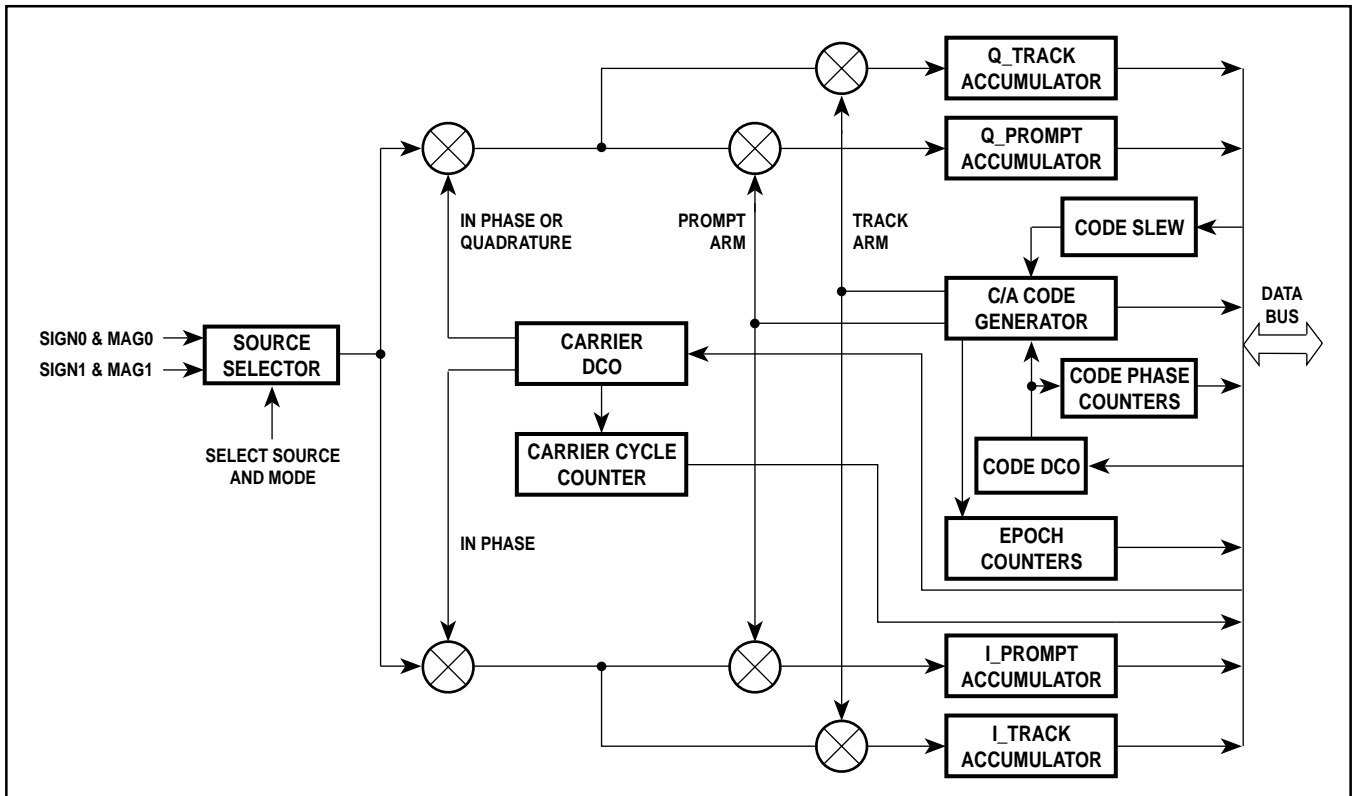
Fig. 9 GP2021 tracking module architecture

Tracking Module

The GP2021 has 12 identical tracking module blocks, one for each channel. Each block (shown in Fig. 9) contains all the components necessary for acquiring and tracking the received signal (code generator, code DCO, carrier DCO, mixers and accumulators). Each block also contains other functional blocks which are used to produce part of the measurement data set (epoch counter and carrier cycle counter).

Code Generator

The Code Generator can generate the following pseudo-random codes:
(i)   GPS satellite C/A codes for PRNs 1 to 32.
(ii) Pseudolite C/A codes for PRNs 33 to 37.
(iii) INMARSAT GIC codes for PRNs 201 to 211

The generated code is supplied to the Prompt and Track arms. The Track arm code can be configured to 4 different modes with respect to the Prompt arm code phase:
(i)   Early: permanently 1/2 chip early (advanced).
(ii) Late: permanently 1/2 chip late (retarded).
(iii) Early-Minus-Late: signed difference of Early and Late.
(iv) Dithered: alternates between Early and Late every 20ms.

Code DCO

The Code DCO clocks the code generator at nominally twice the C/A code rate (2·046MHz). This rate is required to produce the Track arm codes phased by 1/2 chip with respect to the Prompt arm code.

A 26-bit long counter clocked at 40/7MHz (for a 40MHz master clock) provides a resolution of approximately 85·149mHz (42·575mHz at the C/A code rate).

It is programmed via the following registers:
CHx_CODE_DCO_INCR_HIGH: DCO bits 24 to16 and
CHx_CODE_DCO_INCR_LOW: DCO bits 15 to 0.

A control word of 016EA4A9$_{hex}$ gives a C/A code rate of nominally 1·023 MHz.

Code Phase Counters

This block contains counters for determining the code phase of the Prompt arm. At each TIC the counters are sampled and the data is stored in the following registers:

CHx_CODE_PHASE: gives the number of 1/2 chips of code phase in the range 0 to 2046.

CHx_CODE_DCO_PHASE: gives the fractional code phase below 1/2 a chip. It takes values from 0 to 1023 with a resolution of 1/2048 (approximately 0·5 ns or 0·15 m) of a chip.

Epoch Counters

This block contains counters for determining the complete number of code cycles (epochs) on the Prompt arm. At each TIC, the counters are sampled and the data is stored in the CHx_EPOCH_COUNT register, which contains the number of 1 ms (in the range 0 to 19) and 20ms (in the range 0 to 49) epochs. Hence, the Code Phase registers can be combined with the Epoch Count register to give a code time between 0 and 1 second at a resolution of approximately 0·5ns.

Carrier DCO

The carrier DCOs are used to mix the input signal to baseband prior to correlation with the locally generated codes.

A 27-bit long counter clocked at 40/7MHz (for a 40MHz master clock) provides a resolution of approximately 42·575mHz.

The carrier DCOs generate 4 level, 8 phase sinusoids with the sequence over 1 cycle as shown in Fig. 10.

When the input signal is already in complex (I, Q) form the quadrature DCO can be configured to generate an in-phase carrier.

It is programmed via the following two registers:
(i)   CHx_CARRIER_DCO_INCR_HIGH: DCO bits 25 to 16.
(ii) CHx_ CARRIER_DCO_INCR_LOW: DCO bits 15 to 0.

When used with a GP2015 a control word 01F7B1B9$_{hex}$ gives a carrier DCO frequency rate of approximately 1·405397 MHz.

Carrier Cycle Counter

This block contains counters for determining the number (whole and fractional) of in-phase carrier DCO cycles between the last 2 TICs.

CHx_CARRIER_CYCLE_COUNTER_HIGH and
CHx_CARRIER_CYCLE_COUNTER_LOW:

The block provides the number of positive going zero crossings of the in-phase carrier DCO between the last 2 TICs. The counter is 20 bits long.

CHx_CARRIER_DCO_PHASE: gives the in-phase carrier DCO fractional carrier phase sampled at the current TIC. The counter is 10 bits long to give a resolution of 1/1024 (approximately 0·2 mm at L1) of a cycle.

The above registers are used to form the integrated carrier phase measurements.

Accumulators

Four 16-bit accumulators contain the results of the code correlation over the code period of nominally 1ms. This data can be accessed through the following registers:
(i) CHx_I_TRACK: The accumulation from the in-phase Track arm.
(ii) CHx_Q_TRACK: The accumulation from the quadrature Track arm.
(iii) CHx_I_PROMPT: The accumulation from the in-phase Prompt arm.
(iv) CHx_Q_PROMPT: The accumulation from the quadrature Prompt arm.

Status Registers

There are 4 status registers, three associated with the accumulators and one with the measurement data:
(i) ACCUM_STATUS_ A: Amongst other things, this register contains a status bit for each channel which is set when new accumulation data is available for that channel.
(ii) ACCUM_STATUS_B: Amongst other things, this register contains a status bit for each channel which is set when new accumulation data becomes available for that channel before the last accumulation result was read. This gives an indication that the processor is not accessing some or all of the accumulation registers at a fast enough rate.
(iii) ACCUM_STATUS_C: This register contains a status bit for each channel which is set or cleared according to whether the Track arm of that channel is generating Early or Late code when the arm is configured in Dither mode.
(iv) MEAS_STATUS_A: Amongst other things, this register contains a status bit for each channel which is set when new measurement data becomes available for that channel before the last measurement data was read. This gives an indication that the processor is not accessing some or all of the measurement data registers at a fast enough rate.
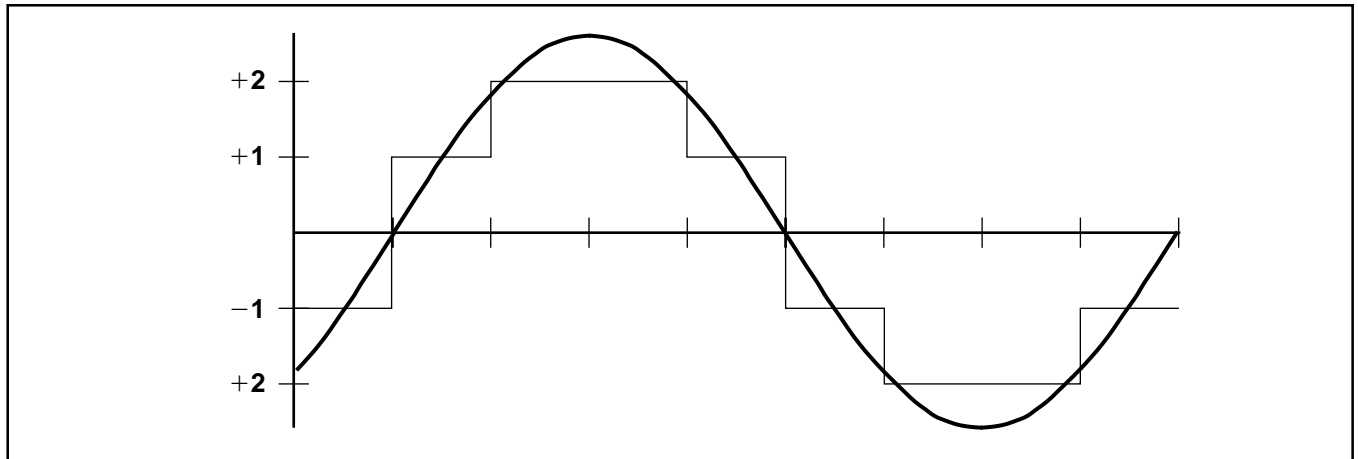

Fig. 10 GP2021 carrier DCO

GPS ARCHITECT HARDWARE AND SOFTWARE OVERVIEW

SYSTEM OVERVIEW

The GPS Architect is a development system intended for Global Positioning System (GPS) receiver design where a GPS receiver function needs to be embedded within OEM products at low cost.

GPS Architect operates as a 12-channel 'All-in-View' GPS receiver. The product is a stand-alone receiver, with all the processing of GPS signals done on a single PCB, with a PC used as a software development tool in conjunction with monitor software (WINMON) which allows the GPS function of the GPS Architect to be monitored and controlled.

This section provides an overview of the component parts of GPS Architect.

GPS Architect comprises a single printed circuit board of overall size 200mm×170mm, which fits into a thin-profile equipment case. The board contains the GP2000 chipset, which includes:

- GP2010 RF front-end (sister product of the GP2015)
- DW9255 35·42MHz SAW with 2MHz bandwidth
- GP2021 12-channel correlator
- ARM 60-B 32-bit RISC microprocessor

Fig. 11 is a block diagram of the circuit used for GPS Architect and the component layout is shown in Fig. 12.

The GPS Architect operates in conjunction with a PC which uses a compiler for the ARM 60-B: the ARM Toolkit. Source code for the GPS Architect can be written in the C programming language, and compiled on the PC. The compiled binary image file produced can be passed along an RS232 connection from the PC to the DEBUG RS232 port on the GPS Architect, and stored in fast SRAM memory, with the help of UMON software which is executed from EPROM. The code execution can be controlled from the PC.

The function of the ARM Toolkit with the GPS Architect is explained in the GPS Architect Data Sheet (DS4605).

One GP2021 UART port is used by the PC-based monitor for receiver control and data display. The second UART port is available for DGPS corrections.

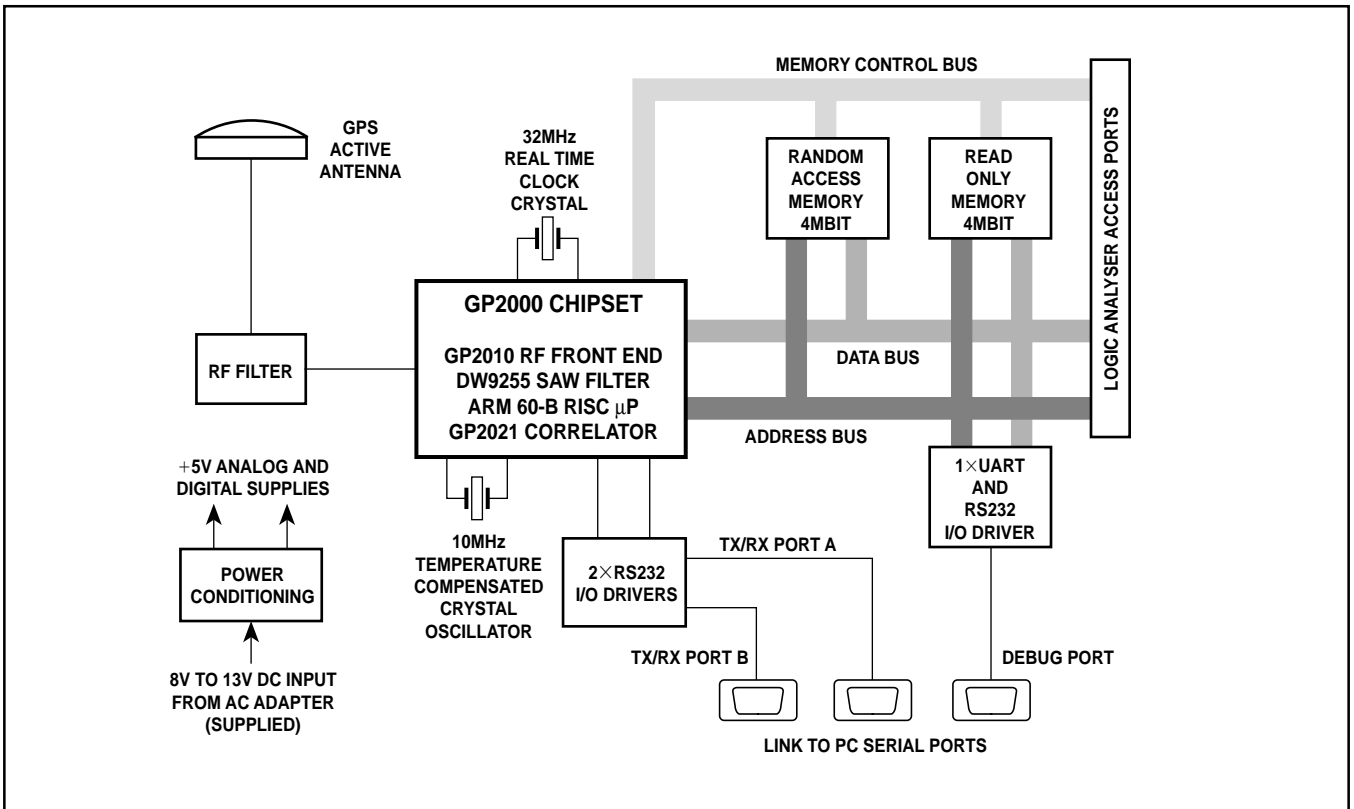Both the ARM Toolkit and the receiver monitor programs are Windows™ based.
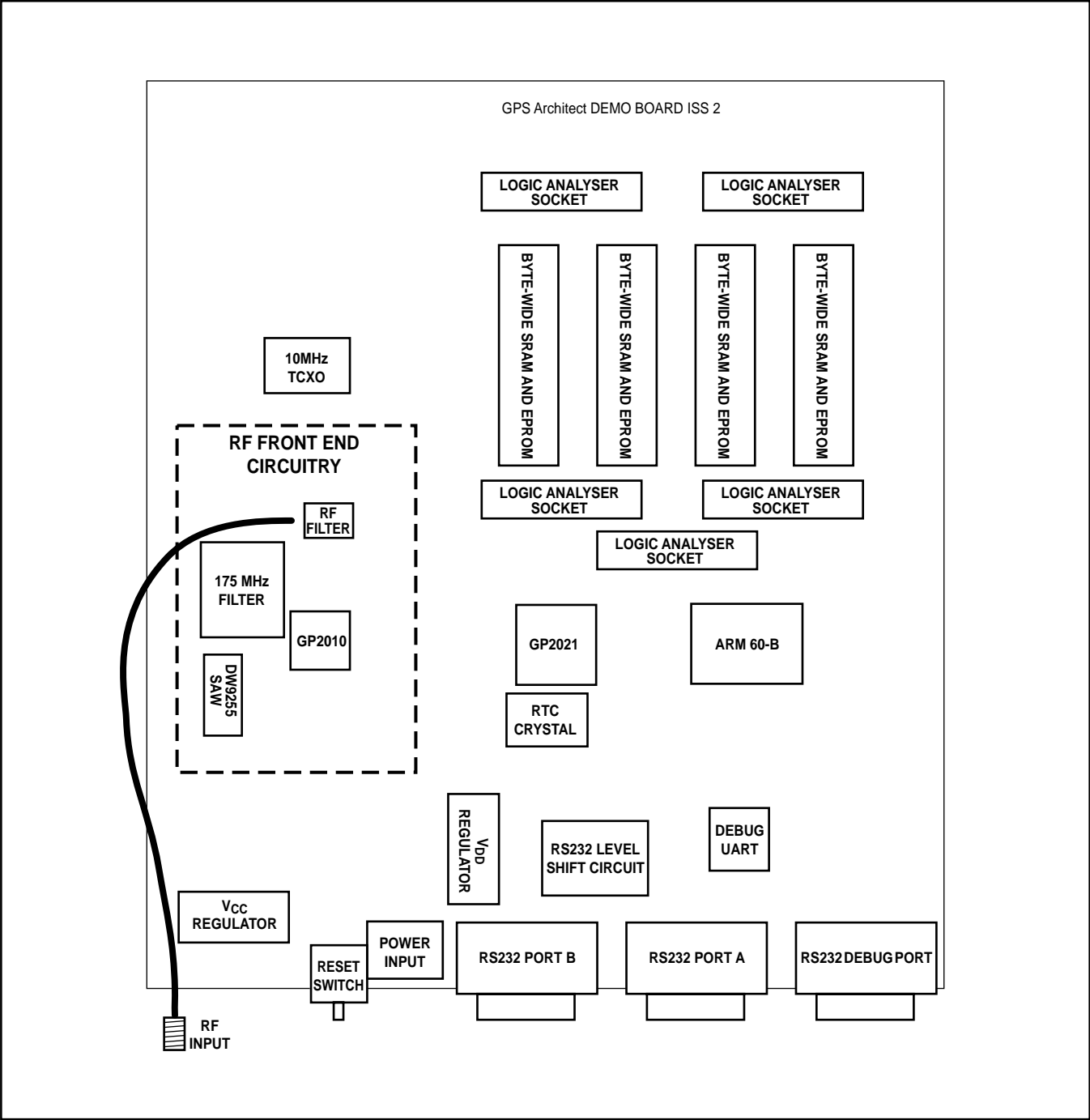

Fig. 11 GPS Architect block diagram

Fig. 12 GPS Architect board layout

SOFTWARE OVERVIEW

The GPS Architect runs a compiled binary image which has been down-loaded from a PC running the ARM Toolkit shell. The ARM Toolkit uses the C programming language for source code for the GPS Architect.

Software Attributes

- 340 KBytes of 'C' source code
- 82 KBytes of 'H' include files
- 8 KBytes of 'S' assembler files
- 157 KBytes executable (40K×32bit)
- Compiles with ARM Toolkit v2.0

Concurrent Tasks

The software performs the following concurrent tasks:

- Read the correlator channel accumulators in the GP2021
- Control the correlator channel tracking loops
- Monitor the loop lock conditions
- Update the receiver position/velocity estimates
- Parse the GPS Data message received from the satellites
- Collect measurement blocks
- Predict satellite positions with respect to observer, based on ephemeris data
- Control satellite selection strategy
- Produce RS232 output data stream
- Check for existence of RTCM SC-104 DGPS data for differential corrections, and parse

Task-Switched Operating System

The GPS Architect software operates under a simple task switching operating system to provide a concurrent structure of operations which lends itself to the requirements of GPS signal processing and software.

The task-switched operating system consists of interrupt and task driven routines. Only one task can be active at any time.

The main source of interrupt is derived from the GP2021 ACCUM_INT signal (505·05μs period default — set to 900·025μs in GPS Architect), and is serviced by the Interrupt Service Routine (ISR). The ISR controls the maintenance of the GPS Architect operating system and processing of the raw GP2021 accumulation data for maintaining the correlator signal tracking loops.

SOFTWARE STRUCTURE

Task Management

GPS Architect has 7 defined tasks and 1 interrupt service routine. New tasks can easily be added by the user.

Tasks can be either active or suspended. Only one task is active at a time. Tasks suspend themselves upon partial or full completion.

Each concurrent software task takes one of two general forms:

1. A task which is to be reactivated at fixed time intervals
2. A task which is to be suspended for a fixed time interval

Task priority detemines which task gets the processor during task conflict. The Main() task is always set to have the lowest priority. Processor retention can be achieved by disabling task-switching or by disabling (masking) GP2021 sourced interrupts.

A task suspension routine operates within the operating system, to suspend a task's operation for a defined number of TICs (1 TIC is 0·0999999s).

Tasks can be added or removed from the software by amending the global Task Control Block structure, TCB. The ordering of the tasks in the declaration of TCB is by task priority.

For each task the TCB data entries are:

- Pointer to task start address
- Pointer to task's stack (initialise as NULL)
- Task stack pointer at start-up (initialise as 0)
- Task stack pointer at suspension (initialise as 0)
- Task name
- Size of task stack

Task Descriptions

The following sections contain descriptions for the GPS Architect software interrupt routine and tasks. An overview of the software concurrent task structure is shown in Fig. 13.

Interrupt Routine

Upon an interrupt from the GP2021 the following actions occur:

- Saving of the current task's context.
- Reading of the new accumulation data.
- Updating of the code, carrier and data demodulation loops.
- Input and output of data for the GP2021 DUART monitor channel (channel B by default).
- Switching of the restore task, if necessary.
- Restoration of the restored task's context.

TTakeMeas()

The TTakeMeas() task's primary function is to collect measurement data from the active channels to be used in the navigation solution. Under appropriate conditions it also performs processing to aid the acquisition and re-acquisition of satellite signals.

TNav()

The TNav() task determines the navigation solution from the data supplied in the channels' measurement blocks. All available data is used in a least-squares solution. Under appropriate conditions a constrained 2-D solution is produced when measurements from only 3 satellites are available.

TDisplay()

The TDisplay() task constructs the output data sentences for the GP2021 DUART monitor channel. The constructed sentences are stored in an output buffer awaiting transmission.

TRTCM()

The TRTCM() task checks and decodes RTCM SC-104 DGPS correction data supplied through the GP2021 DUART port A. Message types 1, 2 and 9 are recognised and used in the navigation solution if valid.

TProcSbf()

The TProcSbf() task decodes and validates the received satellite data message. It also, as appropriate, maintains the satellite position prediction model for the almanac and ephemeris data and performs the initial setting of the receiver clock.

TAlloc()

The TAlloc() task is used to check for received User commands from a PC monitor (e.g. WINMON) and to update the allocation of satellites to channels.

Software Porting Issues

It is likely that all GPS receiver software will contain some host or implementation dependent features. GPS Architect software is no exception. However, to facilitate the porting of GPS Architect code to different processors, such features are kept to a minimum by maximising the use of a high-level programming language (ANSI 'C') and grouping unavoidable dependencies in single modules.

As a consequence of this, and the desire to maintain a high level of generality, scope exists for improvement to the general efficiency and reduction in processor loading of the software. This can be achieved by optimising the software for a particular system (processor, memory structure, etc.), although at the expense of further portability.
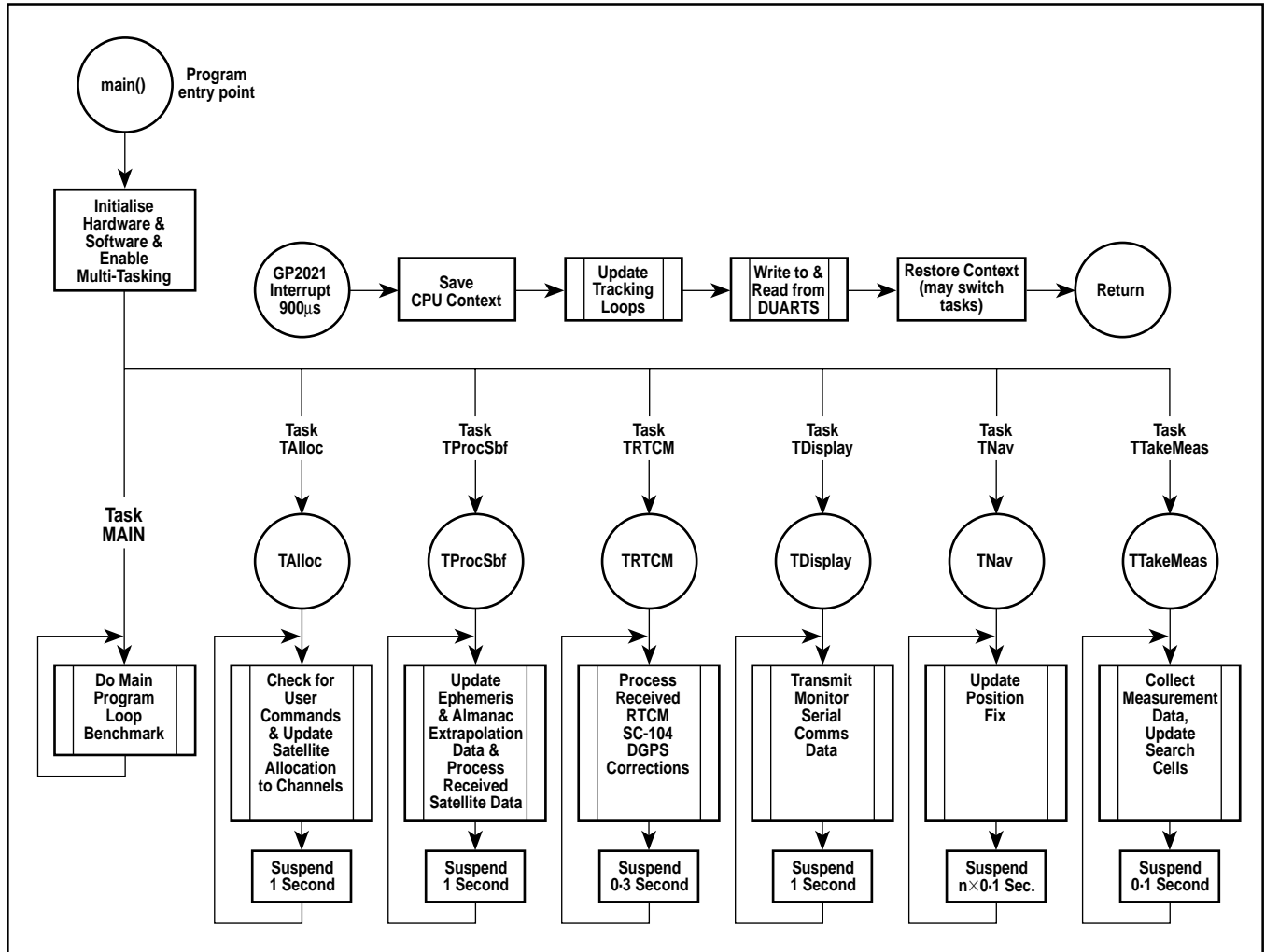

Fig. 13 Concurrent structure of GPS Architect software

GPS ARCHITECT SIGNAL PROCESSING ALGORITHMS

CODE ACQUISITION AND TRACKING

For signal acquisition to occur, both the GP2021 code phase and carrier frequency must match the incoming code phase and carrier frequency to such an extent that the resultant correlation is above the detection threshold.

GPS Architect performs a search in the code phase/carrier frequency plane by searching across all code phases at a series of frequency bins until the signal is detected.

Once signal detection occurs, the code and carrier tracking loops are closed.

Code tracking is achieved using a Phase Locked Loop. Carrier tracking is achieved using a Frequency Locked Loop.

The carrier tracking loop aids the code tracking loop.

Code Search

GPS Architect uses a sliding-replica search for code acquisition.

The GP2021 code DCO is programmed to a slightly higher chip rate than the predicted rate so that the codes 'slide' past each other with time.

The default programmed offset gives a search rate of 0·25 chips per millisecond. Hence, a complete code search (1023 chips) at a given frequency takes about 4 seconds to complete.

The frequency bins are 500Hz wide. Therefore, to search a frequency space of ±10·25kHz takes about 164 seconds to complete.

Code Lock

The current correlation power is given by the sum of the squares of the in-phase and quadrature accumulations.

$$\text{Correlation power} = (I^2 + Q^2)$$

When the instantaneous value of $(I^2 + Q^2)$ exceeds a fixed threshold above the noise floor, signal detection is declared. A filtered version of $(I^2 + Q^2)$ is then monitored for continued code lock.

The accumulator noise floor is determined from the product of the samples from the GP2010 output with the output from GP2021 carrier DCOs.

The effect of code phase error and carrier frequency error on the correlation power level has to be included in the determination of the code lock threshold.

Code Lock Indicator - Noise Floor

The GP2010 has the distribution at its output as given in Table 1:

Level	Percentage of time
+3	15
+1	35
−1	35
−3	15

Table 1

The carrier DCOs have the following distribution over 1 cycle:

$$+2 \quad +2 \quad +1 \quad -1 \quad -2 \quad -2 \quad -1 \quad +1$$

The noise floor for one accumulator $(I^2 + Q^2)$ is given by the mean square of the product of the above two sequences (taking into account the GP2010 distribution) times the number of samples over one millisecond.

$$\begin{aligned}(I^2 + Q^2) &= (22\cdot5 \times 0\cdot3 + 2\cdot5 \times 0\cdot7) \times 40/7 \times 1000 \\ &= 48{,}571\end{aligned}$$

Therefore, the noise floor,

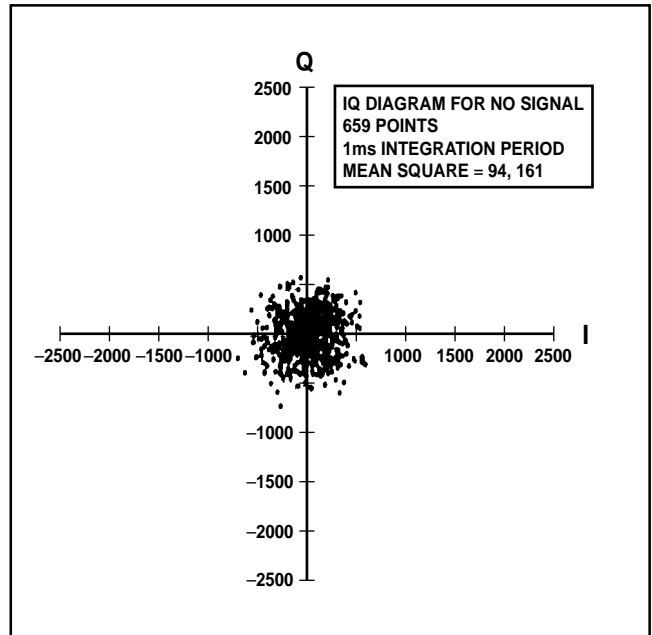$$(I^2 + Q^2) = 2 \times 48{,}751 = 97{,}142$$

Fig. 14 Code lock indicator – noise floor

Code Lock Indicator Acquisition Threshold

The noise floor with no signal present is 97,142. For reliable acquisition, the minimum post-detection SNR will be about 6dB. This corresponds to an acquisition threshold, T_A, of:

$$T_A = 97{,}142 \times 10^{0.6} = 386{,}729$$

The correlation power degrades as a function of a code phase error, Δc chips, by:

$$\text{Correlation loss} = 20\log(1 - \Delta c)$$

Therefore, for $\Delta c = 0\cdot25$ chips, correlation loss = 2·5dB.

The correlation power degrades as a function of a carrier frequency error, Δf Hz, by:

$$\text{Correlation loss} = 20\log\left(\frac{\sin(\pi\Delta fT)}{\pi\Delta fT}\right)$$

Therefore, for $\Delta f = 250$Hz (500Hz/2), correlation loss = 0·9dB

Therefore, including the above losses,

$$T_A = 97{,}142 \times 10^{0.26} = 176{,}769$$

Code Tracking Loop

GPS Architect uses an Early-Minus-Late discriminator for code tracking. The Track arm of the correlator is set 1/2 chip early of the Prompt arm. The code phase error is then given by:

$$EML = (IT^2 + QT^2) - (IP^2 + QP^2)$$

i.e., when the loop is locked the Track arm will be nominally 1/4 chip early and the Prompt arm 1/4 chip late of the actual, but physically non-existent, prompt or on-time code.

The code tracking loop uses a second order Phase Locked Loop. (Note: this can be a first order PLL when used with carrier aiding).

The open-loop transfer function is:

$$Y(s)/X(s) = G(s) = (T_2 s + 1)/T_1 s$$

where $T_1 > T_2 > 0$

or, rearranging:

$$T_1 s \times Y(s) = (T_2 s + 1) \times X(s)$$

Expressing this in the time domain gives:

$$T_1 \times \frac{dy}{dt} = T_2 \times \frac{dx}{dt} + x$$

Over a sample interval ΔT:

$$\frac{dy}{dt} = \frac{(y_i - y_{i-1})}{\Delta T}; \quad \frac{dx}{dt} = \frac{(x_i - x_{i-1})}{\Delta T}$$

therefore:

$$T_1 \times (y_i - y_{i-1}) = T_2 \times (x_i - x_{i-1}) + X_i \Delta T$$

or more conveniently:

$$y_i = y_{i-1} + \left(\frac{T_2}{T_1}\right) \times (x_i - x_{i-1}) + \left(\frac{\Delta T}{T_1}\right) \times x_i$$

The loop is designed so that the factors (T_2/T_1) and $(\Delta T/T_1)$ are powers of 2 to reduce processor loading. T_1 and T_2 are related to the loop characteristics as follows:

Loop natural frequency, $\omega_n = \sqrt{\dfrac{K_\phi \times K_0}{T_1}}$

Damping factor, $\zeta = \left(\dfrac{T_2}{2}\right)\omega_n$

Pull-in range $\Delta\omega_p = \dfrac{4\sqrt{2\zeta\omega_n K_\phi K_0 - \omega_n^2}}{\pi}$

Pull-in time from $\Delta\omega_0$, $T_p = \dfrac{\pi^2}{16} \times \dfrac{\Delta\omega_0^2}{\zeta\omega_n^3}$

Pull-out range, $\Delta\omega_{p0} = 1\cdot8 \times \omega_n \times (\zeta+1)$

Closed loop bandwidth, $W = \omega_n\left(\dfrac{1+4\zeta^2}{8\zeta}\right)$ Hz

K_ϕ and K_0 are the phase error detector gain (the EML discriminator gain, units per radian of code phase error) and the code DCO conversion gain (rad/s per control unit) respectively.

Carrier Acquisition & Tracking
Carrier Tracking Loop

The carrier tracking loop tracks the incoming carrier to produce carrier cycle and carrier phase measurements to smooth the code pseudo-ranges. It also aids the code tracking loop since the ratio of the code to carrier frequency is a constant. Carrier acquisition and tracking is initially achieved using a 4-quadrant frequency discriminator to reduce the frequency error from a few hundred Hertz to a few Hertz.

This is followed by a 2nd order Frequency-Locked Loop (FLL) which has zero steady-state error for a constant rate of change in frequency (acceleration).

FLLs offer superior dynamic performance, robustness and insensitivity to interference over PLLs.

Four-Quadrant Frequency Discriminator

The discriminator correction term is derived by comparing I and Q correlations between successive readings (assuming a data bit transition has not occurred):

$$\Delta I = I_k - I_{k-1}$$
$$\Delta Q = Q_k - Q_{k-1}$$

The choice of correction and sign of application is determined from the current correlations and their respective magnitudes (see Fig. 15).

For $|I_k| > |Q_k|$: IF $I_k > 0$ correction $= \Delta Q$, ELSE correction $= -\Delta Q$
For $|I_k| \leq |Q_k|$: IF $Q_k > 0$ correction $= -\Delta I$, ELSE correction $= \Delta I$
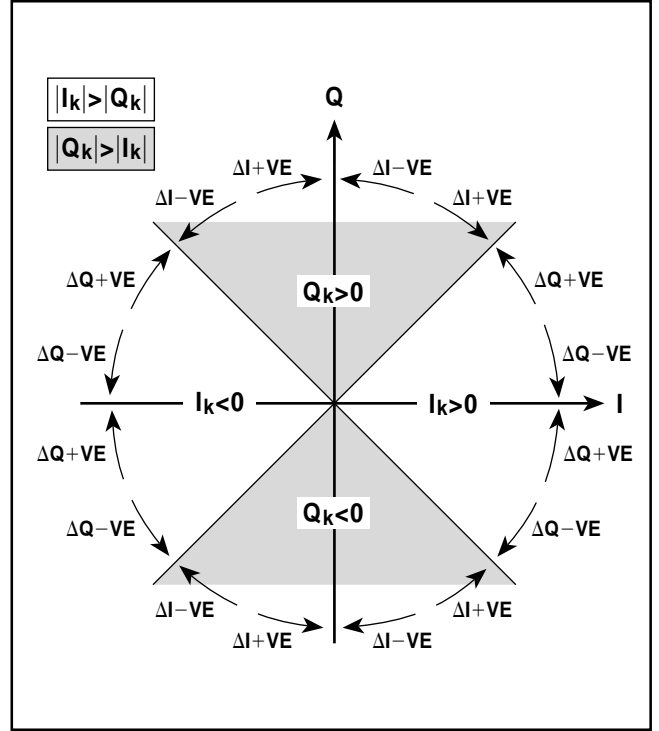

Fig. 15 Four-quadrant frequency discriminator

Frequency-Locked Loop

The discriminator correction term is derived by comparing I and Q correlations between successive readings (assuming a data bit transition has not occurred) using the cross product:

$$f_k = Q_k I_{k-1} - I_k Q_{k-1}$$

For $I_k^2 + Q_k^2 \approx I_{k-1}^2 + Q_{k-1}^2$:

$$f_k = (I_k^2 + Q_k^2)\sin(\phi_k - \phi_{k-1})$$

where ϕ_k and ϕ_{k-1} are the carrier phases at successive readings.

Therefore, for small $\phi_k - \phi_{k-1}$,

$$f_k \approx (I_k^2 + Q_k^2)\Delta\phi,$$

where $\Delta\phi$ is the carrier phase change over 1 millisecond.

Using a discretised second order Jaffe-Rechtin filter of bandwidth B_{LF}, and normalising to the correlation power, the following frequency correction terms can be derived for the estimated carrier frequency:

$$\Delta\omega = \omega_k - \omega_{k-1} = T\dot\omega_k + \sqrt{2\omega_{nF} f_k}$$

$$\omega_{nF} = 1\cdot89 B_{LF}$$

T = sampling interval of 1ms.

The optimum setting for B_{LF} for a given rate of change of acceleration (jerk) is given by:

$$\frac{\ddot\omega}{\omega_{nF}^2} = \frac{0\cdot25}{T}$$

Carrier Lock Indicator

Carrier lock can be monitored by averaging the dot product between correlations:

$$\text{Lock indicator} = I_k I_{k-1} + Q_k Q_{k-1}$$

This quantity averages to zero until the loop locks which in turn drives the cross product, $Q_k I_{k-1} - I_k Q_{k-1}$, to zero. Once locked, the average value of the dot product can be used to estimate the signal power.

15

Data Demodulation

For coherent data demodulation the carrier samples are rotated through the average carrier phase, ϕ_K, as follows:

$$I_{new} = I_k\cos\phi_k + Q_k\sin\phi_k$$
$$Q_{new} = Q_k\cos\phi_k + I_k\sin\phi_k$$

I_{new} can be integrated over the bit period (20ms) to give a representation of the current data bit. To minimise computational loading the functions $\sin()$ and $\cos()$ are determined through look-up tables. Successful (no parity errors) data demodulation occurs for post-correlation SNRs down to approximately 5-6dB.
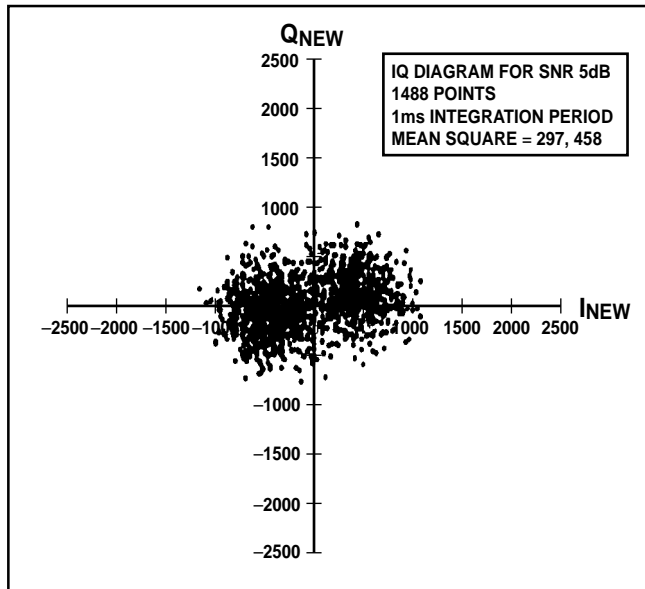

IQ DIAGRAM FOR SNR 5dB
1488 POINTS
1ms INTEGRATION PERIOD
MEAN SQUARE = 297, 458

Fig. 16 Data demodulation with SNR = 5dB

IQ DIAGRAM FOR SNR 15dB
1488 POINTS
1ms INTEGRATION PERIOD
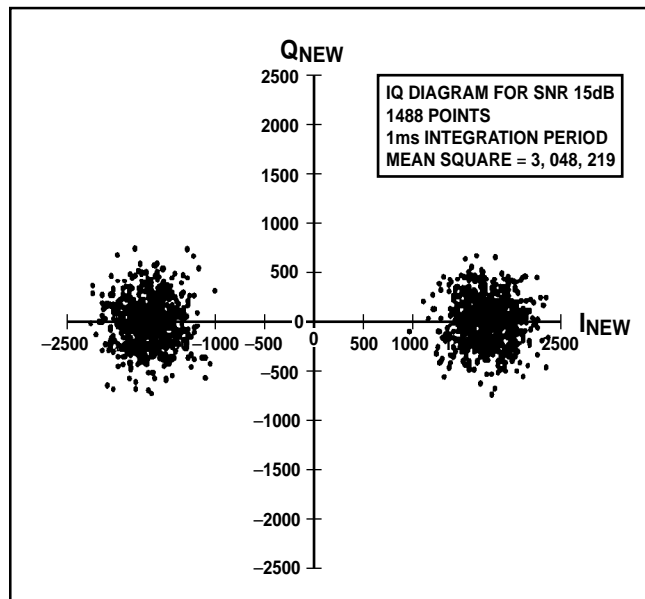MEAN SQUARE = 3, 048, 219

Fig. 17 Data demodulation with SNR = 15dB

Data Bit And Data Frame Synchronisation

Data Bit Synchronisation

The data bit transition is determined by performing a running sum of the in-phase correlations over a bit period at each of the possible twenty 1-millisecond epoch settings. The epoch with the largest integration is declared as the provisional data bit transition. If the best epoch does not change over a 2-second period then bit sync is declared at this position.

The 1ms Epoch Counter is slewed so that its value is zero at the data bit transition, if this is not already the case. The data bit transition position is continually monitored.

Data Frame Synchronisation

Frame sync is achieved by searching for the following parameters contained in the TLM and HOW words across 60 successive data bits (2 words):
(i) TLM preamble (10001011)
(ii) HOW subframe ID (1 to 5)
(iii) HOW zero bits (bits 29 and 30)

If all the above data is found, and the 2 words pass the parity check, then the value of the 20ms Epoch counter is checked. The 20ms Epoch Counter should have a value of zero at the start of each subframe and so a value of 10 at the end of the second word (HOW word). If its value is 10, then frame sync is declared. If it is not 10, then the 20ms Epoch Counter is slewed to its correct value.

Frame sync will then be declared 1 subframe later.

Observation Measurement Block

Measurement Blocks

At each TIC the following data is stored, for each active channel, to be used in the navigator function:

(i) If the channel has valid data
(ii) The satellite PRN assigned to the channel
(iii) The epoch count register (1ms & 20ms epoch counts)
(iv) The code phase register (number of 1/2 chips of code)
(v) The code DCO phase register (the fractional code phase)
(vi) The carrier DCO phase register (the fractional carrier phase)
(vii) The carrier DCO phase register at the last TIC
(viii) The carrier cycle count (whole cycles between TICs)
(ix) The lost lock indicator

Formulation of The Pseudo-Range, Pseudo-Range Rate and Integrated Carrier Phase

Formulation of The Pseudo-Range

At each TIC the pseudo-range is given by the time of reception of the signal minus the time of transmission. This measurement will include the receiver clock bias. Other corrections fo the satellite clock error, Earth rotation plus atmospheric and relativistic effects are applied later.

The time of transmission (modulo 1 second) is determined from the epoch counters and code DCO setting:

Transmit Time = 20×20ms_EPOCH_ COUNT
+1_ms_EPOCH_COUNT
+CODE_DCO_PHASE
−1/4 chip

The code DCO phase includes the number of 1/2 chips plus the fractional phase.

The 1/4 chip term is included because the measurement block data is referenced to the Prompt arm which is set 1/4 chip late in the EML Tracker.

The Reception Time is the receiver clock estimate of GPS System Time at the measurement block TIC. The receiver clock is assumed to be accurate to $\pm 1/2$ second to avoid any ambiguity in the pseudo-range. For all practical systems this will be the case.

Hence, Pseudo-range = Reception Time − Transmission Time (modulo 0·5 seconds).

The pseudo-range is transformed into units of metres.

The pseudo-range rate (including satellite and receiver clock drifts) can be determined from the offset of the carrier DCO from its nominal value:

$$\text{Pseudo-range rate} = -\Delta D \times c / L1$$

where ΔD is the carrier DCO offset from its nominal value, c is the speed of light and $L1$ is 1575·42MHz.

The nominal carrier DCO setting is 88540000/63 or 1450396·825Hz.

Formulation of Integrated Carrier Phase

Integrated carrier phase or integrated Doppler is used to smooth the code phase pseudo-ranges:

$$f_D = \frac{-v \times L_1}{c}$$

where f_D is the Doppler frequency and v the relative velocity. The integral of the Doppler frequency gives a measure of the range change over the integration interval:

$$\int f_D = -\int \frac{v \times L_1}{c}$$

$$f_D = f_{DCO} - f_{nom}$$

where f_{DCO} is the carrier DCO frequency (Doppler shifted) and f_{nom} is the nominal carrier DCO frequency (unshifted). Therefore:

$$\int f_{DCO} - \int f_{nom} = -\int \frac{v \times L_1}{c}$$

$$N_k - T \times f_{nom} = - \frac{L_1 \Delta R}{c}$$

where N_k is the carrier DCO cycle count between TICs, T is the TIC interval and ΔR is the change in range between TICs. Hence,

$$\Delta R = \frac{c \, (T \times f_{nom} - N_k)}{L_1}$$

A running sum of ΔR can then be used to smooth the code pseudo-ranges.

Determination of the Navigation Solution

The basic radial range measurement to satellite i is given by:

$$\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} + ct = R_i$$

where (x, y, z) is the receiver location, (x_i, y_i, z_i) is the satellite position, t is the receiver clock offset and R_i the pseudo-range.

To solve for the 4 unknowns, (x, y, z) and t, 4 equations or pseudo-ranges to 4 satellites are required. Commonly, a linearised version of the range equation is used:

$$\Delta_{Ri} =$$

$$\frac{(x - x_i)}{(R_{ni} - ct_n)} \Delta x + \frac{(y - y_i)}{(R_{ni} - ct_n)} \Delta x + \frac{(z - z_i)}{(R_{ni} - ct_n)} \Delta x + c \Delta t$$

where (x_n, y_n, z_n) and t_n are the nominal (best estimates) of (x, y, z) and t; Δx, Δy, Δz and Δt are the corrections to these estimates; R_{ni} is the nominal pseudo-range measurement and ΔR_i is the difference between the actual and nominal measurement.

The range equations can then be expressed in matrix notation by:

$$Ax = r \text{ or } x = A^{-1} r$$

where A is the solution or design matrix (the direction cosine matrix), x is the receiver position and clock correction vector and r is the pseudo-range measurement difference vector.

More generally, for the overdetermined case (more than 4 satellites), the method of least squares is used:

$$Ax - r = v$$

where v is the vector of residuals. Since the solution matrix is no longer square the generalised or pseudo-inverse must be used:

$$x = (A^T W A)^{-1} A^T Wr$$

where W is the weight matrix.

It can be shown that the optimum value of W is the inverse covariance matrix of the pseudo-ranges. This can be estimated from the satellite URAs (User Range Accuracy) transmitted in the satellite data messages.

If the solution is underdetermined (only 3 measurements), or if the GDOP is above the desired GDOP mask, then altitude aiding is used by adding an extra measurement for an imaginary satellite at the centre of the Earth. The added pseudo-range is equal to the magnitude of the current receiver range vector from the Earth's centre.

Pseudo-range rates are used in the same manner as pseudo-ranges to determine the receiver velocity vector and clock drift.

GPS ARCHITECT PC MONITOR FACILITY – WINMON

The GPS Architect receiver functions are controlled and monitored from a separate PC, running a Windows software package called WINMON. WINMON for the GPS Architect provides a complete operating and monitoring interface for the GPS Architect GPS receiver. The link between the GPS Architect hardware and the PC is RS232 compatible, at 19200 baud, 8 data bits, no parity, and NO flow control.

The WINMON interface provides a number of simple Windows commands, accessed with a mouse or a keyboard, to allow dynamic re-configuration of a number of the key receiver parameters, e.g. number of channels, operation masks, reference position, automatic or user-selection of satellites, loading and saving of satellite almanac files, initialising cold starts etc.

Navigation and receiver status data can be saved to file in comma-delimited text format. GPS Architect can also accept RTCM SC-104 differential corrections data through a second RS232 port.

Screen displays   (Figs 18 to 26)

The following displays can be produced by the WINMON V1.06 software running on a PC. The data in this example was obtained from a GPS Architect GPS receiver running 6.12 receiver software, with RTCM SC-104 differential corrections.

Note that while 12 satellites are in view and locked, the differential beacon providing the DGPS data operates with an elevation mask of 5 degrees. SV25 (channel 10) therefore has no correction data and is excluded from the nav. solution; hence the references to 11 SVs in Fig. 18 and 11 pseudo-ranges in Fig. 19.

The Navigation Data and System Status windows (Figs. 18 and 19) are permanently displayed by WINMON, whereas all the other windows (Figs. 20 to 26) can be switched in and out as required.

```
NAVIGATION DATA                                                    _□✕
Lat  N 51°34.7955'  Spd     0.16  GDOP  1.3  SVs   11  HE 5737840.9  Date 10/02/98
Lon  W  1°46.1583'  Hdg   181.4°  PDOP  1.2  3D Nav     VE     243.3  UTC   12:27:10
Hgt        243.28   ROC    -0.03  VDOP  0.9  DGPS       DO    2164.6  OscErr  -1.38
```

Fig. 18 Navigation data display

```
SYSTEM STATUS                                                      _□✕
3-D NAVIGATION IN PROGRESS (11 PSEUDO-RANGES)
```

Fig. 19 System status display

CH	SV	ELV	AZI	DOPP	NCO	UERE	SF	PRerr	PRRerr	ICPerr	DiffC	LOCK	SNR
1	1	45	289	1101	3264	32	3	1.0	-0.2	0.0	-31.3	CCBF	16.8
2	4	14	70	1589	3749	32	3	1.1	0.3	0.0	39.7	CCBF	17.4
3	5	84	76	-338	1820	32	3	-4.3	0.2	0.0	18.2	CCBF	22.2
4	6	10	190	3761	5924	32	3	-0.1	0.2	0.0	15.2	CCBF	15.4
5	7	5	34	-3248	-1090	32	3	-1.2	0.3	0.0	10.8	CCBF	11.7
6	8	87	281	128	2289	32	3	3.4	0.2	0.0	-43.1	CCBF	20.8
7	9	47	117	-2452	-288	32	3	-1.7	0.1	0.0	10.7	CCBF	18.6
8	21	5	253	-2887	-721	32	3	-2.0	-0.5	0.0	15.7	CCBF	11.4
9	24	6	108	2923	5090	32	3	2.7	-0.5	0.0	21.8	CCBF	8.9
10	25	4	295	3444	5609	32	3	0.0	0.0	0.0	0.0	CCBF	12.2
11	29	11	323	3163	5322	32	3	1.2	-0.1	0.0	-8.3	CCBF	15.9
12	30	51	246	2277	4441	32	3	3.4	-0.2	0.0	6.5	CCBF	20.0

Fig. 20 Channel status display

```
┌──────────────────────────────────────────────────────────────────────────────────┐
│ ▓ SATELLITE SUMMARY                                                     _ □ ✕ │
├──────────────────────────────────────────────────────────────────────────────────┤
│                                                                                  │
│   SV STATUS ELV AZI   DOPP    SV STATUS ELV AZI   DOPP    SV STATUS ELV AZI   DOPP │
│    1        45 289   1088    13       -59 115   -970    25         4 295   3444 │
│    2       -27  69  -2916    14        -3   5   2325    26        -4 161  -3622 │
│    3       -62 251    683    15       -12 333  -2947    27       -53 113  -2348 │
│    4        14  70   1579    16       -15  27   2879    28 NoInfo --- ---   ---- │
│ » 5        84  75   -348    17       -31 196    616    29        11 323   3158 │
│    6        11 190   3760    18       -60  27   1809    30        51 246   2271 │
│    7         5  34  -3253    19       -78 103   -718    31       -65 329  -1375 │
│    8        87 280    116    20 NoSuch --- ---   ----    32 NoSuch --- ---   ---- │
│    9        47 117  -2459    21         5 253  -2891                             │
│   10       -37 125    786    22       -37 248   2977    IONO/UTC                 │
│   11 NoSuch --- ---   ----    23        -4 222  -3088                             │
│   12 NoSuch --- ---   ----    24         6 108   2920                             │
│                                                                                  │
└──────────────────────────────────────────────────────────────────────────────────┘
```

Fig. 21 Satellite summary display

```
┌──────────────────────────────────────────────────────────────────────────────────┐
│ ▓ PROCESSING STATUS                                                     _ □ ✕ │
├──────────────────────────────────────────────────────────────────────────────────┤
│ Accumulations Max Pending:    0    Missed:    20                                 │
│ Measurements                       Missed:    28                                 │
│ Observations  Max Pending:    0    Missed:     0                                 │
│ Subframes     Max Pending:   10    Missed:     0                                 │
│                                                                                  │
│ Channels allocated: 12    Channels in use: 12                                    │
│                                                                                  │
│                                                                                  │
│                                                                                  │
│                                                                                  │
│                                                                                  │
└──────────────────────────────────────────────────────────────────────────────────┘
```

Fig. 22 Processing status display

```
┌──────────────────────────────────────────────────────────────────────────────────┐
│ ▓ OPERATING PARAMETERS                                                  _ □ ✕ │
├──────────────────────────────────────────────────────────────────────────────────┤
│ Current Operating Parameters                                                     │
│                                                                                  │
│ Elevation Mask      :   0.0 degrees                                              │
│ PDOP Mask           :  10.0                                                      │
│ Search Mode         : Highest Elevation Satellites.                              │
│ Altitude Aiding     : AUTO                                                       │
│ Fixed Position      : OFF                                                        │
│ Fix Filter Position Coefficient:   8                                             │
│ Fix Filter Velocity Coefficient:   2                                             │
│                                                                                  │
│                                                                                  │
│                                                                                  │
└──────────────────────────────────────────────────────────────────────────────────┘
```
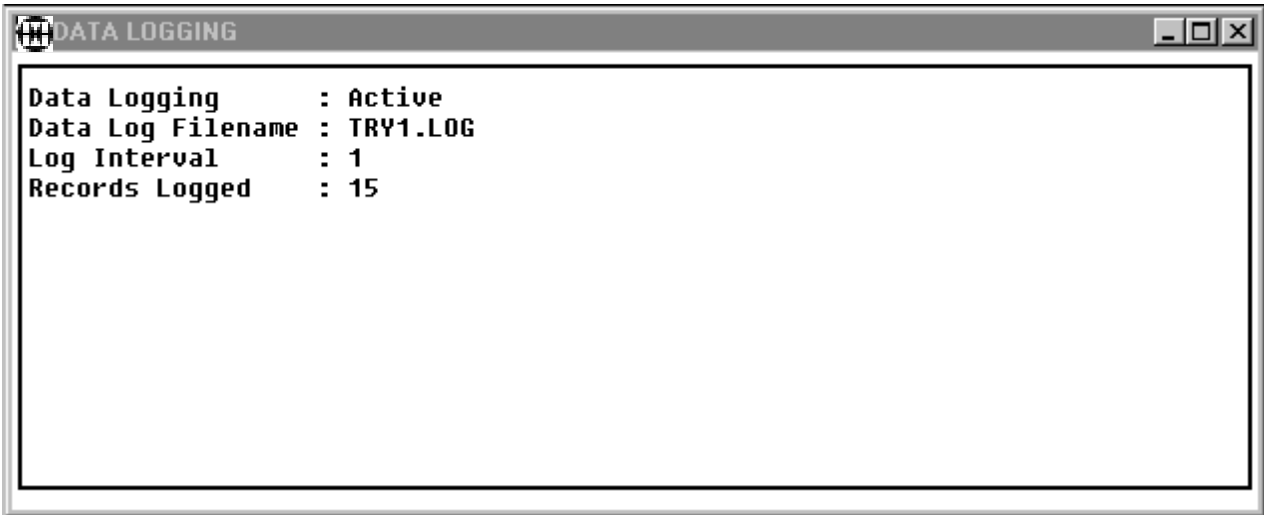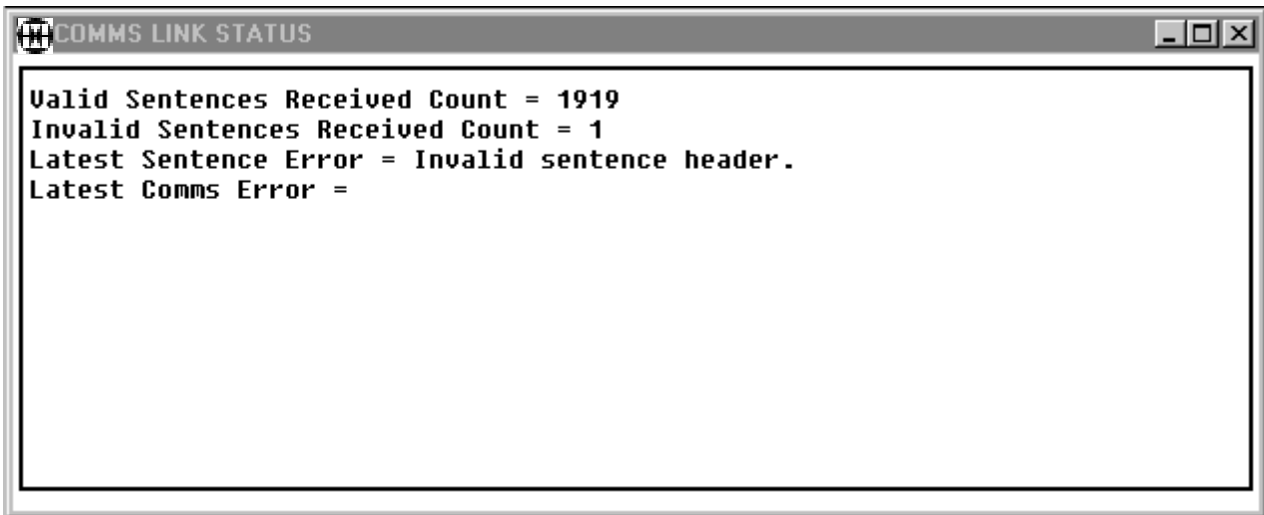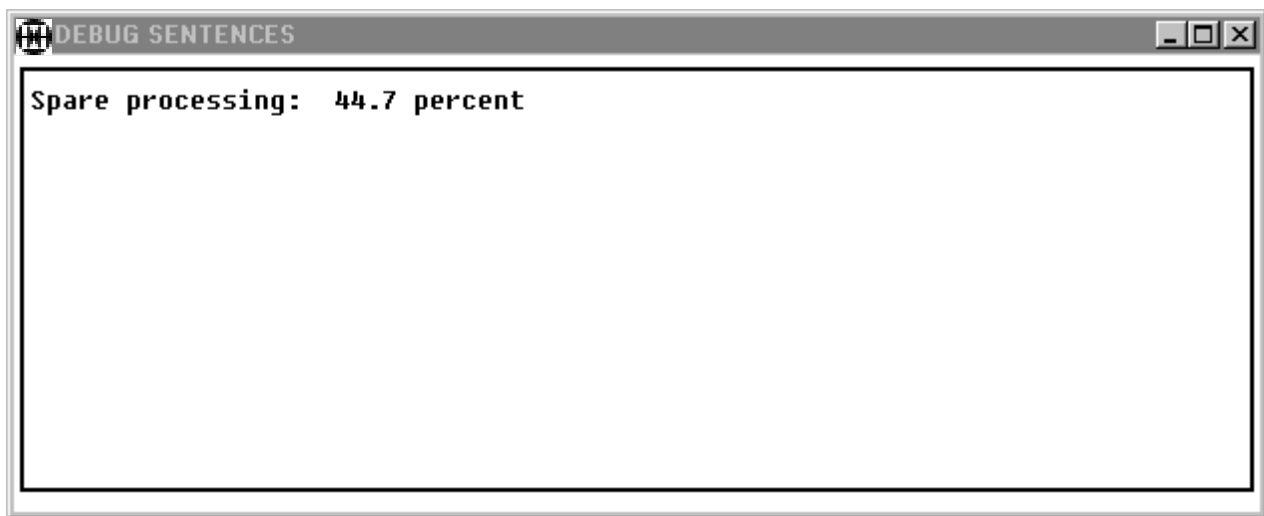
Fig. 23 Operating parameters display

```
DATA LOGGING                                          _ □ ×

Data Logging        : Active
Data Log Filename : TRY1.LOG
Log Interval        : 1
Records Logged      : 15
```

Fig. 24 Data logging status display

```
COMMS LINK STATUS                                     _ □ ×

Valid Sentences Received Count = 1919
Invalid Sentences Received Count = 1
Latest Sentence Error = Invalid sentence header.
Latest Comms Error =
```

Fig. 25 Comms link status display

```
DEBUG SENTENCES                                       _ □ ×

Spare processing:   44.7 percent
```

Fig. 26 Debug sentences display, configured to show spare processing capacity of the ARM 60-B microprocessor

Performance Parameters

SV selection strategy

The SV selection strategy (i.e. which satellites the receiver tries to acquire) can be complex, and not always successful. If the number of receiver channels is less than the number of visible satellites (or satellites above the receiver elevation mask) then decisions have to be made about which satellites to acquire.

A common criterion for selection is those satellites which give the smallest dilution of precision in the navigation fix. This is fine if the receiver antenna has full sky visibility but if there are obstructions shielding the satellites from the antenna (particularly relevant in dynamic applications) then either some complex software is required to swap satellites in and out of the acquisition loop or worse still, the receiver may stick to the 'optimum' constellation until either the satellites eventually become visible or the best constellation changes. The number of satellites above the horizon is obviously a position and time dependent quantity.

However, for reasonable latitudes (say below $\pm60°$), the average number of visible satellites will be about 8 and on occasions will increase to 10, 11 or even 12. Hence, having as many receiver channels as visible satellites means that all satellites can be tracked (or at least acquisition can be attempted) without the need for a complex or time consuming selection strategy.

In GPS Architect, when sufficient data is available, the initial satellite-to-channel allocation is based in descending elevation angle and this is updated as satellites rise and set. The ability to track all visible satellites ('all-in-view') also adds to the position accuracy and availability.

Signal Acquisition Times

The initial signal acquistion time for a particular satellite in GPS Architect is determined by how many frequency bins have to be searched to find the signal. The number of bins is determined by the bin width and the maximum frequency excursions due to the satellite receiver relative motion and the receiver clock error. For a static receiver the maximum Doppler excursion is about ±5kHz.

The reference oscillator used with GPS Architect is a Rakon™ TXO4010 with the following stability:

Temperature : ±2.5 ppm $(-30$ to $+75°C)$
Supply voltage : ±0.2 ppm $(+5V\pm5\%)$
Ageing : ±1.0 ppm/year

Assuming a total reference oscillator error of ±4 ppm i.e. ±6.3kHz at $L1$ plus the Doppler error of ±5.0kHz gives a total frequency excursion of about ±11.3kHz.

For the GPS Architect frequency bin width of 500Hz a total of 45 bins exists. For a 4-second bin search time this gives a total of 180 seconds to cover the complete range. However, if the clock error is known (but no estimate of receiver or satellite positions) then this decreases the search window to ±5.0kHz, the number of bins to 20 and the search time to 80 seconds (worst case).

If the satellite almanacs, the reference clock error and a reasonable estimate of the receiver position and current time (errors of 1.3Hz per km of position error and 0.9Hz per second of time error) are available then it should be possible to constrain the search within 1 to 3 bins, giving an acquisition time of 4 to 12 seconds (Hot Start).

Bit And Frame Sync Times

Following signal acquisition, the next step is bit and frame sync. A minimum of 2 seconds dwell time is required for bit sync to the data transitions. To achieve frame sync the TLM and HOW words need to be successfully received and the 20ms Epoch Counter set.

If valid data is being logged (continuous reception and no parity errors), then TLM and HOW reception will take between 1.2 and 7.2 seconds. If the 20ms Epoch Counter is slewed then frame sync will be delayed 6 seconds.

Hence, to go from code/carrier lock to bit and frame sync will normally take between about 3 and 15 seconds.

Signal Re-Acquisition Times

When a signal is lost, the tracking loops for the channel concerned go into a coast mode. In this mode the code and carrier DCOs are left free running at their current values for the duration of the maximum coasting interval (user configurable) or until the code and carrier detection thresholds are exceeded. If the maximum coasting interval is exceeded, the channel returns to the search mode. If the signal reappears (e.g., after being temporarily obstructed by a building) within the maximum coasting interval then signal reacquisition typically occurs within 1 second.

Time-To-First-Fix

TTFF is dominated by how much initial information GPS Architect has. Note: The satellite almanacs and ephemerides can be read from file and the clock frequency error and initial position estimate supplied via command line.

If no information is available (i.e., position estimate, time estimate, reference oscillator error, satellite almanacs and ephemerides) then a sky search occurs, selecting satellites sequentially from their PRN ordering. In this scenario, TTFF is typically 3 to 6 minutes* (Cold Start).

If GPS Architect has satellite almanacs, a reasonable time and position estimate and a reference oscillator error estimate (the usual scenario) then TTFF is typically less than 1 minute.* Note that it can take 30 seconds to receive an ephemeris (Warm Start).

If GPS Architect has the above information and valid (less than 4 hours old) satellite ephemerides then TTFF is typically less than 30 seconds* (Hot Start).

*These figures assume a static receiver and unobstructed sky visibility.

Condition	Cold start	Warm start	Hot start
Reference oscillator error	4ppm Temperature, voltage and ageing	Known to within ±0.1ppm	Known to within ±0.1ppm
Satellite almanacs	Unknown	1 week old	N/A
Satellite ephemerides	Unknown	Unknown	< 4 hours old
Initial position estimate	Unknown	< 100km	< 100km
Initial time estimate	Unknown	< 5 minutes	< 5 minutes

Table 2 Additional information on TTFF conditions

STATIC NAVIGATION

The GPS Architect Global Positioning Development System has been developed from GPS Builder-2, a PC-based GPS development system (no longer available). Two GPS Builder-2 systems have been tested side-by-side to illustrate the improved accuracy in position fixes that results from using a 12-channel receiver rather than a 4-channel receiver.

Fig. 27 shows a GPS Builder-2 position fix scatter plot where

10 satellites are included in the solution. Fig. 28 shows position fixes over the same period but for a GPS Builder-2 system using only the 'best' 4 satellites. A comparison of these plots clearly demonstrates the benefit to position fix accuracy of 'all-in-view' used by GPS Architect.

Fig. 29 shows the sort of position fix accuracy that can be achieved when S/A is inactive (no DGPS corrections applied). Fig. 30 shows the number of satellites tracked during this period.
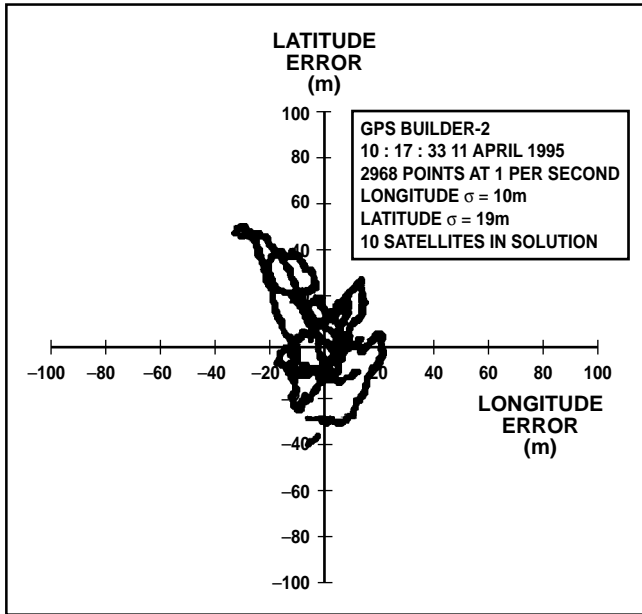
Fig. 27 Static navigation using GPS Builder-2, 12 channels, S/A active
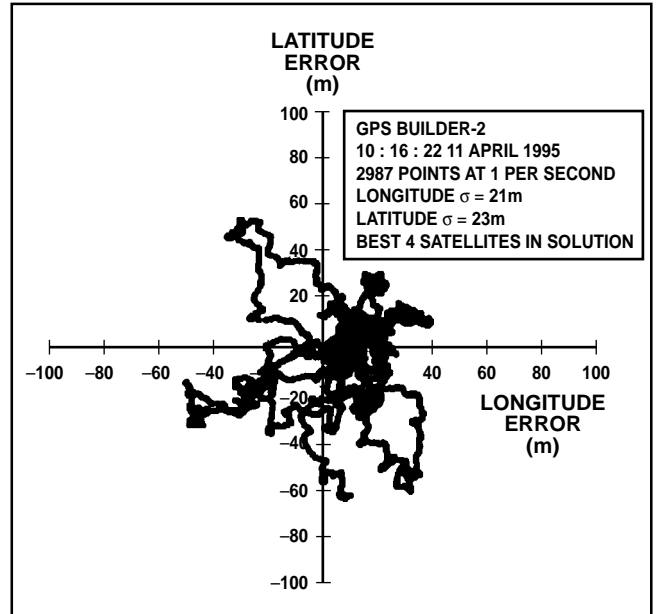
Fig. 28 Static navigation using GPS Builder-2, 4 channels, S/A active
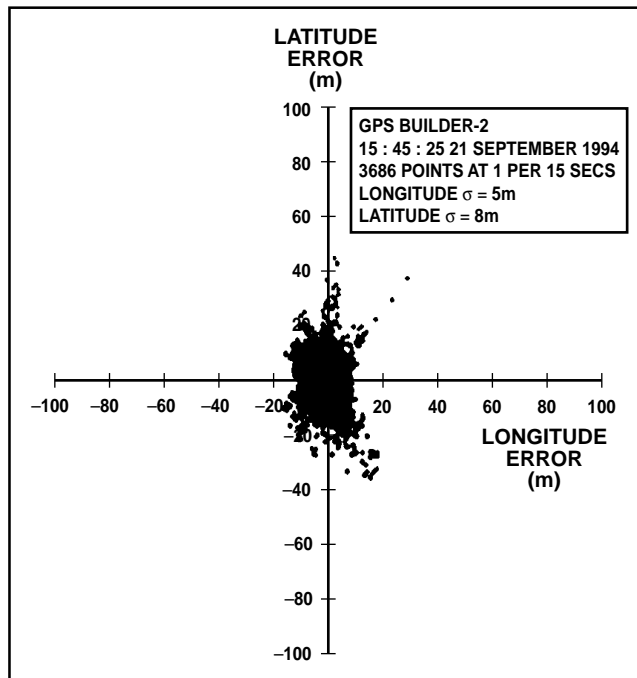
Fig. 29 Static navigation, S/A inactive
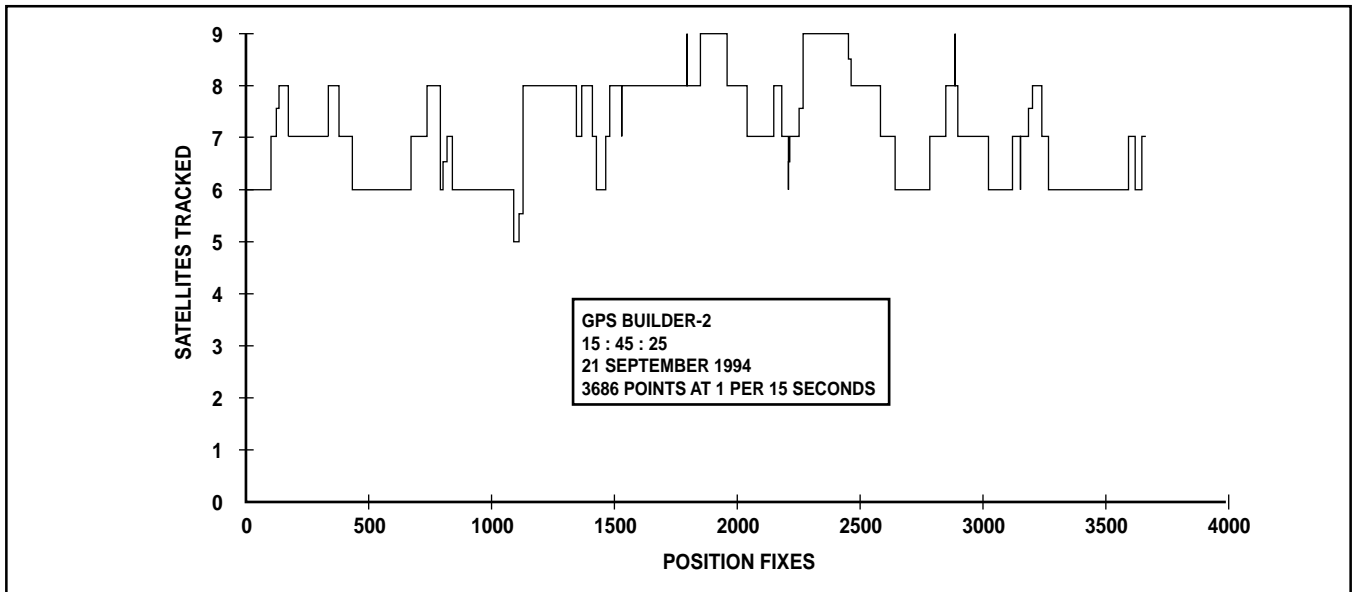
Fig. 30 Number of satellites tracked

GPS ARCHITECT SOFTWARE ENHANCEMENTS

GPS Architect is a generalised solution to the GPS receiver problem. To maintain a high level of clarity, generality and adaptability, no significant attempts have been made to target the solution for one particular application area. This provides numerous opportunities for customisation to user-specific applications.

Such areas include:

(i) Acquisition times. Acquisition/re-acquisition times can be improved by using both the Prompt and Track arms of the correlator in the acquisition search. This could involve stepping the code phase as opposed to a sliding search.

In addition, bit and frame sync times can be shortened by using all the available information to determine epoch settings.

(ii) Navigation filter. GPS Architect uses a very simple, bandwidth-definable, low pass navigation filter. This has NOT been optimised for any particular application and needs to be 'tailored' by the user accordingly.

(iii) Additional aiding. In applications where GPS is not sufficient as the sole means of navigation information then the navigation solution could contain aiding from external sources (e.g. gyroscopic sensor).

23

GLOSSARY

AGC	Automatic Gain Control		**NMEA**	National Marine Electronics Association
Almanac	A low precision ephemeris		**NVM**	Non-Volatile Memory
ASCII	American Standard Code for Information Interchange		**PDOP**	Position Dilution Of Precision
			PLD	Programmable Logic Device
Bin	Carrier frequency step during signal acquisition		**PLL**	Phase Locked Loop
C/A Code	Coarse/Acquisition code		**PRN**	Pseudo-Random Noise
Chip	One bit of satellite code (C/A code has 1023 chips)		**PROM**	Programmable Read-Only Memory
			RAM	Random Access Memory
CW	Continuous Wave		**RISC**	Reduced Instruction Set Computer
DCO	Digitally Controlled Oscillator		**RINEX**	Receiver-Independent Exchange format
DGPS	Differential Global Positioning System		**ROM**	Read-Only Memory
EML	Early Minus Late		**RTC**	Real Time Clock
Ephemeris	Orbital parameter set for high precision satellite position prediction		**RTCM**	Radio Technical Commission for Maritime Services
Epoch	Code repetition interval (C/A code epoch is 1ms)		**S/A**	Selective Availability
EPROM	Erasable Programmable Read-Only Memory		**SAW filter**	Surface Acoustic Wave filter
E²PROM	Electrically Erasable Programmable Read-Only Memory		**SV**	Space Vehicle, i.e. satellite
			TCXO	Temperature Compensated Crystal Oscillator
FIFO	First-In, First-Out register		**TIC**	A signal of programmable period which is used to sample and latch tracking module measurement data
FLL	Frequency Locked Loop			
GDOP	Geometric Dilution Of Precision			
GLONASS	Global Orbiting Navigation Satellite System		**TLM**	Telemetry
GPS	Global Positioning System		**TTFF**	Time To First Fix
HDOP	Horizontal Dilution Of Precision		**UART**	Universal Asynchronous Receiver/Transmitter
HOW	Hand-Over Word		**URA**	User Range Accuracy
IF	Intermediate Frequency		**UTC**	Universal Time Co-ordinated
ISA	Industry Standard Architecture		**VCO**	Voltage Controlled Oscillator
LNA	Low Noise Amplifier		**VDOP**	Vertical Dilution Of Precision

The information in this Designer's Guide was originally presented as part of the Navtech Seminars programme. Our thanks go to Navtech Seminars.

MITEL SEMICONDUCTOR