Implementing RAM Functions

in FLEX 10K Devices

November 1995, ver. 1 Application Note 52

Introduction

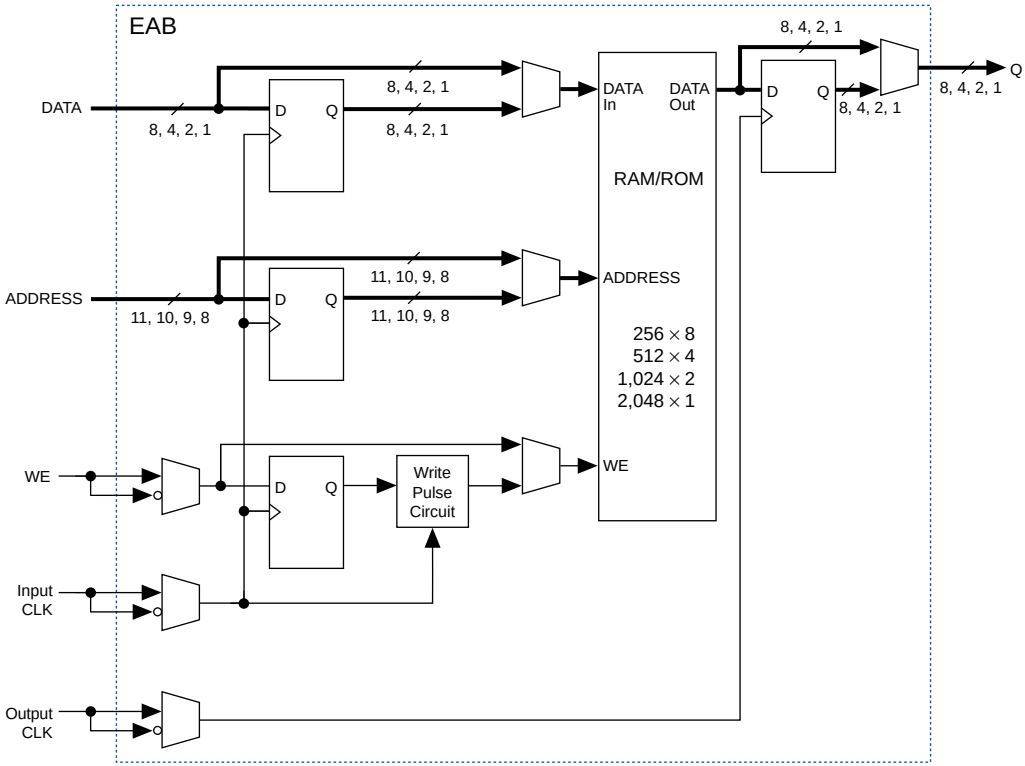
The Altera FLEX 10K family provides the first programmable logic devices (PLDs) that contain an embedded array. The embedded array is composed of a series of embedded array blocks (EABs) that can efficiently implement complex custom functions, including RAM. On-board RAM is useful for designers who require memory in their designs but do not wish to use an additional memory device because of speed requirements, I/O pin conservation, or printed circuit board (PCB) space limitations.

This application brief discusses how to implement synchronous or asynchronous blocks of RAM in FLEX 10K designs.

EAB Architecture

Each FLEX 10K EAB contains 2,048 bits of RAM with a DATA bus up to 8 bits wide and an ADDRESS bus up to 11 bits wide. Unlike the distributed RAM in field-programmable gate arrays (FPGAs), the FLEX 10K EAB is designed to ensure predictable, easy-to-use timing. The EAB Write Enable signal (WE) can be synchronized with the input Clock, or it can be asynchronous. The EAB also contains input, output, and address registers that can be used to synchronize a design. EAB outputs can be registered or combinatorial. Registered outputs can be used to pipeline a design, increasing system performance. Figure 1 shows the EAB block diagram.

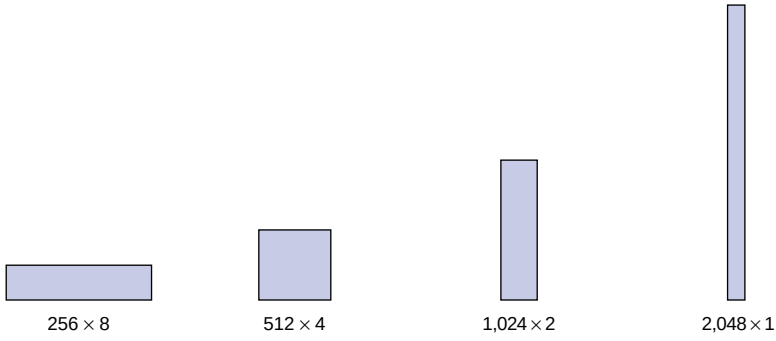
Figure 1. EAB Block Diagram



Using EABs

The EAB RAM size is flexible; therefore, you are not limited to one RAM configuration such as 2,048 words of 1 bit each. Figure 2 shows a graphical representation of the flexible RAM sizes. The width of the DATA and ADDRESS buses varies with the size of the RAM. You can configure an EAB for any size with standard EDA tools or with Altera's MAX+PLUS II development system.

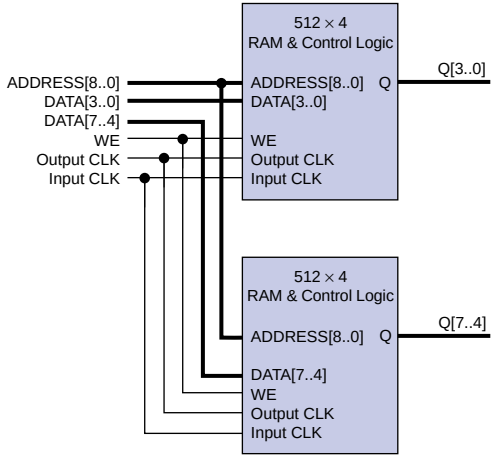
Figure 2. EAB Memory Configurations



Cascading EABs for Wider RAM

The MAX+PLUS II software automatically cascades EABs to implement larger blocks of RAM for designs that require blocks of RAM wider than an EAB configuration (see Figure 3). Cascading FLEX 10K EABs does not require additional logic; therefore, cascaded and non-cascaded EABs have the same access time.

Figure 3. Cascaded EABs

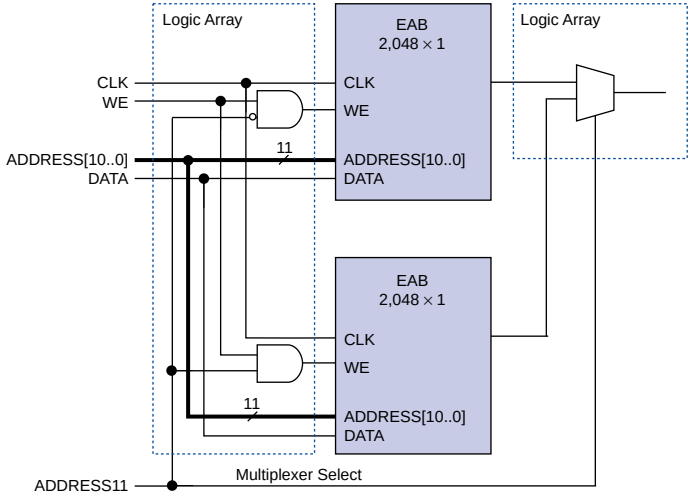


Unlike FLEX 10K EABs, distributed RAM in FPGAs only has predictable access times for small, individual RAM blocks, such as 16 × 1. As the size of the FPGA RAM is increased, additional RAM blocks are needed and the access times become significantly slower. For example, a typical 256 × 8 RAM block in an FPGA may require a 33-ns access time and use a significant number of device resources in addition to RAM.

Multiplexing EABs for Deeper RAM

Each EAB can address 2,048 words with 11 address lines. To address more than 2,048 words, the MAX+PLUS II software automatically multiplexes the outputs of multiple EABs and uses the multiplexer select line as an additional address line. In Figure 4, two EABs—each configured for a block of 2,048 × 1 RAM—are multiplexed to create a block of 4,096 × 1 RAM. Multiplexing EABs to create blocks of RAM deeper than 2,048 words requires additional logic on the outputs, which causes a small additional delay.

Figure 4. Multiplexed EABs



Synchronous RAM

When using a FLEX 10K EAB for synchronous RAM, the DATA and ADDRESS signals are registered in the EAB. When you use an EAB for synchronous RAM, all control signal timing, including the WE signal, is implemented in the EAB. Generating the WE signal within the EAB eliminates potential glitches that can corrupt data. When a high WE signal

is clocked into the EAB, circuitry inside the EAB generates a write pulse that meets setup and hold times for the DATA and ADDRESS inputs. Because the WE signal allows ADDRESS inputs to change while WE is high, you do not have to de-assert WE to write on consecutive Clock cycles. The WE signal automatically pulses high on each Clock cycle.

Asynchronous RAM

You can configure the FLEX 10K EAB for asynchronous RAM using the following guidelines:

- The WE signal must be timed to avoid glitches that can unintentionally overwrite the RAM.
- The WE signal must meet the setup and hold times for DATA and ADDRESS.
- The ADDRESS inputs cannot be changed while WE is high.

Emulating ROM

To configure the FLEX 10K EAB to emulate ROM, you must define the contents of the EAB with a Hexadecimal (Intel-format) File (**.hex**) or an Altera Memory Initialization File (**.mif**). Because the EAB is not write-protected, the start-up data can be reconfigured during device operation, which makes the EAB more flexible than true ROM.

Software Support

You can create designs for FLEX 10K devices using functions from the library of parameterized modules (LPM) with MAX+PLUS II and standard EDA tools, or you can use non-LPM-based Synopsys or Viewlogic logic synthesis tools.

MAX+PLUS II & Standard EDA Tools

MAX+PLUS II and standard EDA tools (from vendors such as Mentor Graphics, Intergraph, Viewlogic, and Cadence) take advantage of the LPM standard, a fast-emerging industry standard. Each module in the library has parameters, i.e., variable attributes, that permit you to represent a wide variety of logic functions.

The LPM parameters are defined in the design entry tool. For example, a graphic editor might display a window for entering parameters when an LPM symbol is chosen. The chosen parameters are then saved for synthesis. In text editors, the LPM function is referenced in the design file and the parameters are set when the function is used.

For example, you can use EDA tools to create a block of RAM and specify the parameters for the width (LPM_WIDTH) and address width (LPM_WIDTHAD) of the RAM. The function and the parameters are passed to the MAX+PLUS II software via an EDIF netlist file. MAX+PLUS II synthesizes the design for the FLEX 10K architecture, automatically cascading EABs to form wider RAM, implementing registered inputs for synchronous RAM, or using additional resources from the FLEX 10K logic array to implement RAM deeper than 2,048 words.

VHDL

With Mentor Graphics' Autologic II and Cadence's Synergy, you can use VHDL to declare LPM entities, such as RAM, with generics that allow parameters to be passed down from upper-level files. You can specify generics with GENERIC MAP aspects in an upper-level file, and synthesize your design with a third-party synthesis tool. Then, you can pass the synthesized design with the LPM functions and parameters to the MAX+PLUS II software via an EDIF netlist file. See Figure 5 for an example of RAM in top-level VHDL code.

Figure 5. Example of RAM in VHDL

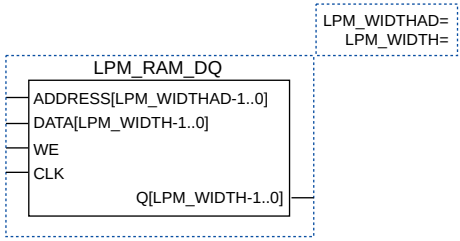
```
ARCHITECTURE <architecture_name> OF <entity_name> IS

BEGIN

ul: lpm_ram_dq GENERIC MAP(lpm_width, lpm_widthad)
   PORT MAP(<value>, <value>)

END <architecture_name>
```

The lpm_width parameter is the word width and the lpm_widthad parameter is the address width. The PORT MAP statement lists the top-level nodes that are connected to the LPM ports.

MAX+PLUS II

The Altera MAX+PLUS II development system supports the LPM standard. You can enter an LPM function, such as LPM_RAM_DQ, using the MAX+PLUS II Graphic Editor as shown in Figure 6.

Figure 6. Example of LPM_RAM_DQ in the MAX+PLUS II Graphic Editor



You can also enter LPM functions in the MAX+PLUS II software using the Altera Hardware Description Language (AHDL) as shown in Figure 7.

Figure 7. Example of LPM_RAM_DQ in AHDL

```
<RAM_name>:LPM_RAM_DQ WITH
(
      lpm_width=<value>-- word width
      lpm_widthad=<value>-- address width
)
```

In addition to LPM functions, the MAX+PLUS II software supports specialized functions, such as a cycle-shared FIFO function (CSFIFO) and a cycle-shared dual-port RAM function (CSDPRAM), that take advantage of EAB architectural features not supported by the LPM.

Synopsys & Viewlogic Tools

You can efficiently use the FLEX 10K EAB in VHDL designs with Synopsys or Viewlogic tools and the Altera software utility **genmem**, which is provided with workstation versions of MAX+PLUS II. The **genmem** utility generates a VHDL hollow-body symbol, including a port description and symbol, and allows you to specify the parameters that define the RAM functionality. You can then instantiate the hollow-body symbol in an upper level VHDL design. Synopsys or Viewlogic tools synthesize the design and pass the RAM symbol to the MAX+PLUS II software for synthesis into the device. Including memory functions in a FLEX 10K design with the **genmem** utility produces the same design efficiency as entering the memory with LPM functions.

Conclusion

You can use LPM functions with MAX+PLUS II or EDA tools, or you can use the **genmem** software utility with Synopsys or Viewlogic tools to efficiently implement powerful megafunctions, including RAM, in FLEX 10K EABs. EABs offer the features and functionality that let you implement RAM functions with minimal effort and give you the flexibility to create the size and functionality you require.

2610 Orchard Parkway
San Jose, CA 95134-2020
(408) 894-7000
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 894-7104
Literature Services:
(408) 894-7144