Figure 8.20  Simulated waveforms for a synchronous memory.

Both the XS95 and XS40 Board implementations will work the same way. Some examples of the actions that can be performed with the synchronous memory are as follows:

<table>
<thead>
<tr>
<th>Action</th>
<th>XSPORT Command Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write location 1 (01)</td>
<td>XSPORT 01110100</td>
</tr>
<tr>
<td></td>
<td>XSPORT 01110101</td>
</tr>
<tr>
<td>Write location 3 (11)</td>
<td>XSPORT 10101100</td>
</tr>
<tr>
<td></td>
<td>XSPORT 10101101</td>
</tr>
<tr>
<td>Read location 0 (00)</td>
<td>XSPORT 0011</td>
</tr>
<tr>
<td>Read location 2 (10)</td>
<td>XSPORT 1011</td>
</tr>
</tbody>
</table>

The MEM2 design uses 51 macrocells of an XC95108 CPLD. Since the LED decoder uses 7 macrocells, the 4×4 memory must use the other 51 – 7 = 44 macrocells. This is a large fraction of the total resources of the CPLD. Obviously, you cannot build a very large memory using the techniques shown in this or the previous section.

Using Built-In Memory

To allow designers to use larger amounts of memory in their designs, the XC4000 series of FPGAs is equipped with internal RAMs. Since the LUTs in FPGAs are essentially RAMs which store the logic truth-tables (refer back to Figure 2.6), it is reasonable to allow the designer to use these RAMs for general-purpose data storage. Each CLB in an XC4000 FPGA has 2 LUTs that can be used as RAMs (Figure 2.7). Each 4-input LUT has room to store \(2^4=16\) bits, so a single CLB can store 32 bits of data. The data can be organized as a 16×2 RAM, a 32×1 RAM, or dual 16×1 RAMs.
(Unfortunately, the XC9500 series of CPLDs does not offer the option of using macrocells as either logic or memory. So the material in this section is irrelevant to those using the XS95 Board. Sorry.)

Listing 8.5 shows some VHDL code which uses the RAM16X4 RAM module that is constructed from two 16x2 built-in RAMs. The COMPONENT declaration for the module is on lines 20–28 and was determined by looking into the RAM16X4.XNF file found in the directory \$ilinx\Synth\ilinx\macros\Xc4000e\v6_xnf. The RAM module has 4-bit input and output buses, a 4-bit address bus, and an active-high write-enable input.

The RAM module is instantiated on lines 52–58. Since we are only building a 4x4 memory, we only need two address bits. Therefore, on line 54 the lower two address inputs of the module are connected to the address bus. As they are not needed, the upper two address bits are connected to the gnd signal which is held at logic 0 on line 49.

Since we are targeting the XS40 Board with this design, we need to get some of our inputs from the mode pins of the XC4005XL FPGA. This is done on lines 43–46. The inputs from the mode pins are concatenated with the 2-bit input bus to create the 4-bit inb bus. The inb bus connects to the RAM module on line 55. The remaining output, and write-enable ports of the RAM module are connected to their counterpart signals in the mem3 design (lines 56 and 57).

Figure 8.21 shows the simulated waveforms for the MEM3 design. Four writes were made to the memory by applying an address and data value and then placing a positive pulse on the WE input. During the interval 0–110 ns the hex values of 0xC, 0x3, 0x5, and 0xA were written to memory locations 0, 1, 2, and 3, respectively. Then the values were read back out of these locations between 120 ns and 160 ns. The value read from each location matches the data that was written to it, so the memory seems to be working.

The user-constraint file for the MEM3 design is as follows:

```
NET we   LOC=P44; # B0 argument of XSPORT
# NET oe LOC=P45; # OE is not used with the internal memories
NET a<0> LOC=P46; # B2 argument of XSPORT
NET a<1> LOC=P47; # B3 argument of XSPORT
NET inp<0> LOC=P48; # B4 argument of XSPORT
NET inp<1> LOC=P49; # B5 argument of XSPORT
# NET inp<2> LOC=P32; # B6 argument of XSPORT
# NET inp<3> LOC=P34; # B7 argument of XSPORT
NET s<0> LOC=P25; # S0 LED segment
NET s<1> LOC=P26; # S1 LED segment
NET s<2> LOC=P24; # S2 LED segment
NET s<3> LOC=P20; # S3 LED segment
NET s<4> LOC=P23; # S4 LED segment
```
Listing 8.5  VHDL version of a 4-bit wide asynchronous memory constructed using a built-in FPGA memory.

```vhdl
001- -- Using Built-in Memory of the XC4000 FPGAs
002-
003- LIBRARY IEEE,xse;
004- USE IEEE.std_logic_1164.ALL;
005- USE xse.led.ALL;
006-
007- ENTITY mem3 IS
008- PORT
009- (  
010-   we: IN STD_LOGIC; -- active-high write-enable control input
011-   a: IN STD_LOGIC_VECTOR (1 DOWNTO 0);-- 2-bit address bus
012-   inp: IN STD_LOGIC_VECTOR (1 DOWNTO 0);-- 2-bit input data bus
013-   s: OUT STD_LOGIC_VECTOR (6 DOWNTO 0)-- drivers for LED
014- );
015- END mem3;
016-
017-
018- ARCHITECTURE mem3_arch OF mem3 IS
019-
020- COMPONENT ram16x4 -- interface for the RAM16X4.XNF component
021- PORT
022- (  
023-   A0,A1,A2,A3: IN STD_LOGIC;-- 4-bit address bus
024-   D0,D1,D2,D3: IN STD_LOGIC;-- 4-bit input data bus
025-   WE: IN STD_LOGIC; -- active-high write-enable
026-   O0,O1,O2,O3: OUT STD_LOGIC-- 4-bit output data bus
027- );
028- END COMPONENT;
029-
030- SIGNAL o: STD_LOGIC_VECTOR(3 DOWNTO 0); -- intermediate output bus
031-
032- -- components/signals for input from the special-purpose mode pins
033- COMPONENT md0 PORT(MD0: OUT STD_LOGIC); END COMPONENT;
034- COMPONENT md2 PORT(MD2: OUT STD_LOGIC); END COMPONENT;
035- COMPONENT ibuf PORT(I: IN STD_LOGIC; O: OUT STD_LOGIC); END COMPONENT;
036- SIGNAL MD0in, MD2in, MD0out, MD2out: STD_LOGIC;
037- SIGNAL inb: STD_LOGIC_VECTOR (3 DOWNTO 0);
038-
039- SIGNAL gnd: STD_LOGIC; -- signal held at logic 0
040-
041- BEGIN
```
Listing 8.5 VHDL version of a 4-bit wide asynchronous memory constructed using a built-in FPGA memory. (Cont'd.)

042-
043- u0: md0 PORT MAP(MDO=>MD0in); -- connect the mode inputs to
044- u1: md2 PORT MAP(MD2=>MD2in); -- the bus feeding the RAM module
045- u2: IBUF PORT MAP(I=>MD0in, O=>MD0out);
046- u3: IBUF PORT MAP(I=>MD2in, O=>MD2out);
047- inb <= MD2out & MD0out & inp(1 DOWNTO 0); -- concatenate bits into bus
048-
049- gnd <= '0'; -- connect signal to logic 0;
050-
051- -- connect the RAM16X4 component I/O to the I/O of this module
052- u4: ram16x4 PORT MAP
053-  (  
054-     A0=>a(0),A1=>a(1),A2=>gnd,A3=>gnd, -- connect address bus
055-     D0=>inb(0),D1=>inb(1),D2=>inb(2),D3=>inb(3), -- connect input bus
056-     Q0=>o(0),Q1=>o(1),Q2=>o(2),Q3=>o(3), -- connect output bus
057-     WE=>we
058-  );
059-
060- u5: ledcdc PORT MAP(d=>o,s=>s); -- show output hex digit on LED
061- END mem3_arch;
062-

Figure 8.21 A simulation of the VHDL design that uses the built-in asynchronous memory of the XC4005XL FPGA.
NET s<5> LOC=P18;   # S5 LED segment
NET s<6> LOC=P19;   # S6 LED segment

Examples of the actions that can be performed with this memory are quite similar to what we had with the asynchronous memory we built from D flip-flops (except that the write-enable is inverted):

<table>
<thead>
<tr>
<th>Action</th>
<th>XSPORT Command Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write location 1 (01) with 7 (0111)</td>
<td>XSPORT 01110110</td>
</tr>
<tr>
<td></td>
<td>XSPORT 01110111</td>
</tr>
<tr>
<td></td>
<td>XSPORT 01110110</td>
</tr>
<tr>
<td>Write location 3 (11) with 10 (1010)</td>
<td>XSPORT 10101110</td>
</tr>
<tr>
<td></td>
<td>XSPORT 10101111</td>
</tr>
<tr>
<td></td>
<td>XSPORT 10101110</td>
</tr>
<tr>
<td>Read location 0 (00)</td>
<td>XSPORT 0010</td>
</tr>
<tr>
<td>Read location 2 (10)</td>
<td>XSPORT 1010</td>
</tr>
</tbody>
</table>

In addition to asynchronous RAMs, the XC4000 series also offers synchronous RAMs and dual-port RAMs. Figure 8.22 gives an example of using an internal 16x2 dual-port synchronous RAM in a schematic. The RAM was instantiated by selecting the RAM16x2D symbol from the SC Symbols window of the schematic editor. It is equipped with a normal address bus (A3..A0) that controls which memory location will be written or read using the INP1..INP0 input bus and SPO1..SPO0 output buses, respectively. In addition, a second read-only port is provided that lets you enter an address through DPRA3..DPRA0 and read the contents of the memory location through output bus DPO1..DPO0. (There are not enough pins on the PC parallel port to drive all these address bits, so the upper two bits of each address were attached to ground in the schematic.)

The constraint file for the MEM4 design follows. Note that each 2-bit output port of the memory is applied to unique pair of LED segments so that we can observe the values output by both independent ports.

NET CLK LOC=P44;   # B0 argument of XSPORT
NET WE  LOC=P45;    # B1 argument of XSPORT
NET INP0 LOC=P46;   # B2 argument of XSPORT
NET INP1 LOC=P47;   # B3 argument of XSPORT
NET A0  LOC=P48;    # B4 argument of XSPORT
NET A1  LOC=P49;    # B5 argument of XSPORT
# NET DPRA0 LOC=P32; # B6 argument of XSPORT
# NET DPRA1 LOC=P34; # B7 argument of XSPORT
NET S1  LOC=P26;    # S1 LED segment (for SPO0)
NET S2  LOC=P24;    # S2 LED segment (for SPO1)
Figure 8.22 Using the built-in synchronous dual-port memory of the XC4005XL FPGA.

NET S4 LOC=P23;  # S4 LED segment (for DPO0)
NET S5 LOC=P18;  # S5 LED segment (for DPO1)

The following command sequences perform the indicated operations:

<table>
<thead>
<tr>
<th>Action</th>
<th>XSPORT Command Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write location 1 (01) with 3 (11) while reading from location 2 (10)</td>
<td>XSPORT 10011110</td>
</tr>
<tr>
<td></td>
<td>XSPORT 10011111</td>
</tr>
<tr>
<td>Write location 0 (00) with 2 (10) while reading from location 1 (01)</td>
<td>XSPORT 01001010</td>
</tr>
<tr>
<td></td>
<td>XSPORT 01001011</td>
</tr>
<tr>
<td>Read locations 0 (00) and 1 (01)</td>
<td>XSPORT 00010001</td>
</tr>
<tr>
<td>Read location 3 (11) and 2 (10)</td>
<td>XSPORT 11100001</td>
</tr>
</tbody>
</table>

Using the internal RAMs, the asynchronous memory consumes only 13 LUTs (most of which are used in the LED decoder) while the dual-port synchronous design takes up 4 LUTs. This is quite a reduction in LUT usage when compared to the flip-flop-based designs of the previous sections.

Using a Memory Chip

Even the internal RAMs of the XC4000 FPGAs will not suffice when you need a large amount of memory. Connecting external memory chips to the pins of an