International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 15 (2019) pp. 3325-3329

Design and Implementation of SRAM Controller on Reconfigurable Platform

Arun Tigadi, Dr Hansraj Guhilot

Visvesvaraya Technological University, India, Department of Electrconics and Communication, K.L.E Dr M.S.Sheshgiri College of Engineering and Technology, Angol Main Road, Udyambag, Belagavi, Karnataka -590008, India.

[2] Shivaji University, India, Principal, K.C.College of Engineering and Management Studies and Research Mith Bunder Road, Near Sadguru Garden, Kopri, Thane (East) - 400603, India.

Abstract

The processors and other electronic applications are getting faster and faster day by day and the need for large quantity of data at high speed is increasing, but it has become difficult to provide the data at such high speed. To overcome this hazard a memory controller is required. This paper describes static random access memory(SRAM) controller for reading and writing the data on to the memory at high speed. The design uses finite state machine(FSM) architecture that is developed for testing of this algorithm. The tool used to simulate this design is Xilinx ISE design suit. The hardware used to synthesize this design is Papilio duo loader which consist of a SRAM chip of 512k and a sparton6 FPGA.

Keywords: FPGA; SRAM; Spartan6; FSM

I. INTRODUCTION

SRAM is a type of a memory that holds data in static form i.e, as long as the memory has power. The data is lost once the power is turned off. SRAM is more faster in speed as compared to other to other types of RAM's. Because of the speed being faster these are more expensive then other RAM's. The main advantage of SRAM is that it does not required to be periodically refreshed. These are generally used in smaller applications such as CPU cache memory and hard drive buffer's.

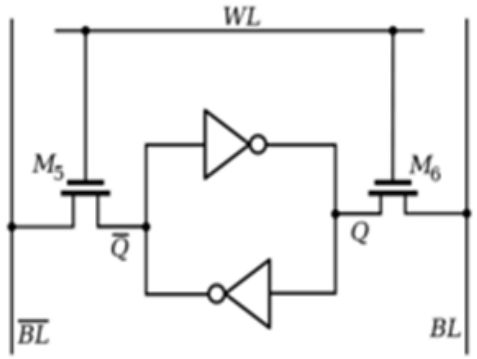


Fig.1. Basic memory cell.

As shown in figure 1 represents the basic memory cell of SRAM which includes six transistors from M1 to M6. Transistors from M1 to M4 are in the form of logic gates and it forms a crossed coupled inverter. BL is a bit line and WL is the word line. When logic 0 is given as the input to the first inverter then the output of the first inverter will be logic 1 which will be the input to the second inverter and the output will be logic 0. This forms a loop which creates a stable state for logic 0. Similarly a stable state is created for logic 1[1].

II. FEATURES OF SRAM

Speed of SRAM as we know that SRAM's are more faster than other RAM's we can easily compare it with DRAM's. The fastest DRAM's which are presently available in market still requires five to ten processor clock cycle to access the first bit of data. The SRAM's can operate at a processor speed of 250MHZ and a clock cycle time as of the microprocessor. Now if we see the density of SRAM it tells us about the compactness of SRAM. SRAM's have low density as compared to DRAM's. SRAM's store less memory per chip where as DRAM's store more memory per chip. If DRAM is holding a data of 64MB then the largest SRAM's can hold a data of only 16MB. SRAM's are volatile in nature that is they can make sudden changes like the data is lost a soon as the power is turned off this nature is called as volatile. As we discussed that SRAM's are expensive because the circuit is complex it requires six transistors this is the reason it requires more area and thus the cost increases. The power consumption of SRAM is comparatively less because refresh is not required[2].

III. BASIC ARCHITECTURE OF SRAM

Figure 2 represents the basic block diagram of SRAM.

It consist of address lines from A0 to A18 a 19 bit of address and data lines from I/O0 to I/O7 a 8 bit of data. It consist of only one port through which the data is written as well as read. The row decoder is used to decode the address and the column I/O to find the block of memory array. Memory array stores the data in the form of rows and columns but in this

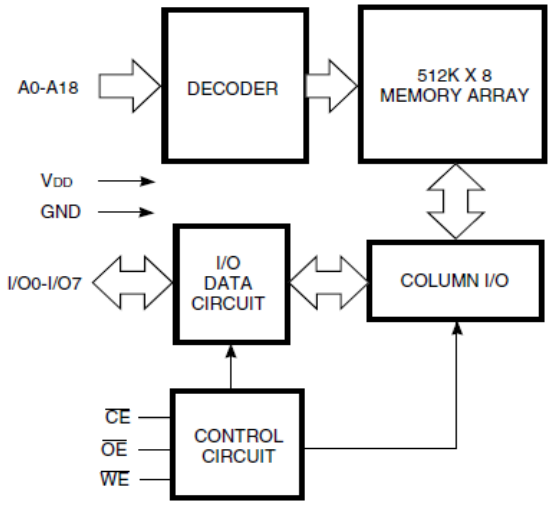


Fig. 2. Basic architecture of SRAM.

case the memory array is in a stack like structure which consist of 512 rows and a single column. Each row consist of 8 bit of data stored into it. There are three control signals.

- Chip enable(CE)
- Output enable(OE)
- Write enable(WE)

CE is the chip enable pin which selects a particular chip and enables it. When CE is active on the chip that chip will respond to read and write request. OE is output enable which is used to read signal from the chip that is it is only used for reading. It indicates that you want the chip to output the data for reading. WE is the write enable signal to the chip on receiving the WE signal the SRAM writes the given data onto the given address. All of these three pins are active low on SRAM chips.

Mode	$\overline{WE}$	$\overline{CE}$	$\overline{OE}$	I/O Operation
Not Selected (Power-down)	X	H	X	High-Z
Output Disabled	H	L	H	High-Z
Read	H	L	L	DOUT
Write	L	L	X	DIN

Fig. 3. Truth table of SRAM.

Figure 3 represents the truth table of SRAM. There are four modes of operations when the power is down the chip enable is high and the operation is high impedance and when the output is disabled write enable and output enable are high where as chip enable is low the operation is high impedance. The last two modes are read and write where in when the chip is selected i.e, when CE and OE are low the operation will be dout i.e, the data will be read from the address and when WE and CE both are low the operation is din i.e, the data will be written onto the given address[3].

IV. SRAM OPERATIONS

As we know that there are two modes of operations read mode and write mode. Now let us first see the write mode of operation. Figure 4 represents the timing diagram for write operation. To write into SRAM when valid address is present and the control signals CE and WE are active low the data din is written onto the given 19 bit of address. At first the address will be setup and then the write operation takes place after writing it will hold the data for some amount of time and the it goes into idle state. Figure 5 represents the timing parameters used in write operation.

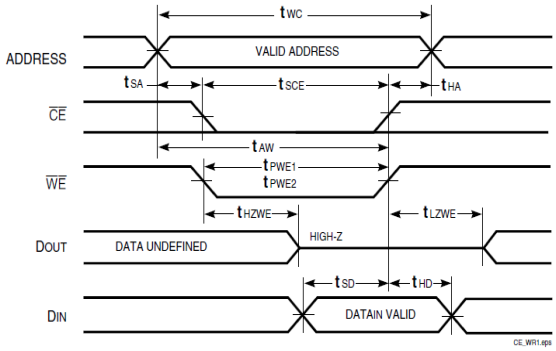


Fig. 4. Write cycle.

Symbol	Parameter
t_{WC}	Write Cycle Time
t_{SCE}	$\overline{CE}$ to Write End
t_{AW}	Address Setup Time to Write End
t_{HA}	Address Hold from Write End
t_{SA}	Address Setup Time
t_{PWE1}	WE Pulse Width (OE = HIGH)
t_{PWE2}	$\overline{WE}$ Pulse Width ($\overline{OE}$ = LOW)
t_{SD}	Data Setup to Write End
t_{HD}	Data Hold from Write End
t_{HZWE}[3]	WE LOW to High-Z Output
t_{LZWE}[3]	WE HIGH to Low-Z Output

Fig. 5. Timing parameters of write cycle.

Next comes the read mode of operation. Figure 6 represents the read cycle. When the valid address is present and CE and OE are active low the data written onto the address will be read at dout. The data cannot be read immediately as soon as CE and OE becomes low it requires some amount of time to fetch the data from the particular address. At first the address will be set up and then the read operation takes place after reading it will hold the data for some amount of time and then it goes into idle state[2].

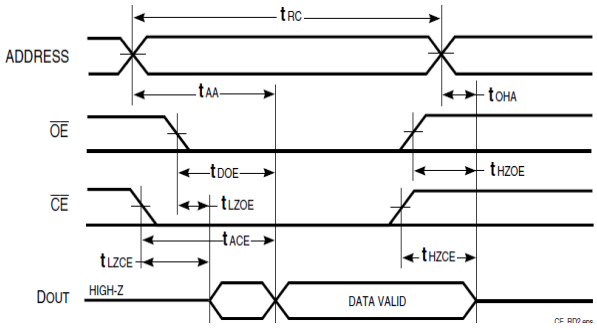


Fig. 6. Read cycle.

Symbol	Parameter
t_{RC}	Read Cycle Time
t_{AA}	Address Access Time
t_{OHA}	Output Hold Time
t_{ACE}	CE Access Time
t_{DOE}	OE Access Time
t_{HZOE}(2)	OE to High-Z Output
t_{LZOE}(2)	OE to Low-Z Output
t_{HZCE}(2	CE to High-Z Output
t_{LZCE}(2)	CE to Low-Z Output

Fig. 7. Timing parameters of read cycle.

Figure 7 represents the timing parameters used for the read operation.

V. SRAM STATE MACHINES

Now, let us see the state machines for SRAM figure 8 represents the state machine for write mode of operation

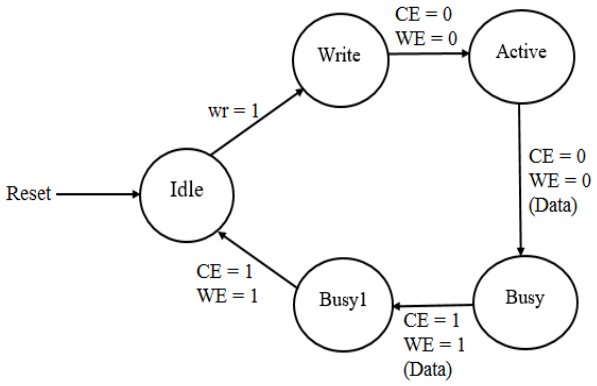
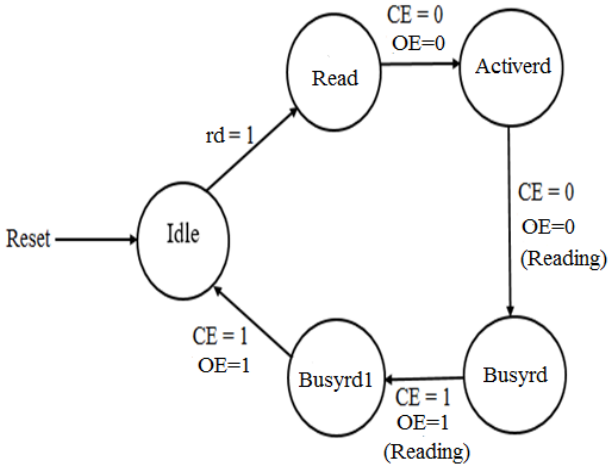


Fig. 8. SRAM write state machine.

If the signal wr=1 then it will be in write state else it will be in idle state. If chip enable and write enable both are zero then the state will be active state and busy state. But in active state it won't write the data and in busy state it starts to write the data onto the given address. If the chip enable and write enable are equal to one it will be writing for some amount of time that state is called as busy1 state and when the address becomes invalid it enters into idle state.[4]

Figure 9 represents the state machine for read mode of operation. When the signal rd=1 then it will be in read state else it will be in idle state. If the chip enable and output enable are equal to zero then the state will be activerd and busyrd state. But in activerd state it won't be reading the data and in busyrd state it will start to read the data from the given address. If the chip enable and output enable are equal to one then also it will be still reading the data for some amount of time that state is called as busyrd1 and then when the address becomes invalid it enters into idle state.



The signals wr and rd are user defined.

VI. RESULTS AND OBERVATIONS

Figure 10 represents simulation result of SRAM which consist of three control signals CE,OE and WE and a 19 bit of address and 8 bit of data. The address goes on incrementing as well the data goes on increasing. The data is written multiple times on to the same address. As we can see from the simulation result that as soon as CE and WE becomes low and address is present the data will be written on to that address. The OE comes into picture for reading the data. The data can be read only when you synthesis the program. The LED signal is used for reading the data on hardware. Figure 11 shows the power analysis report.

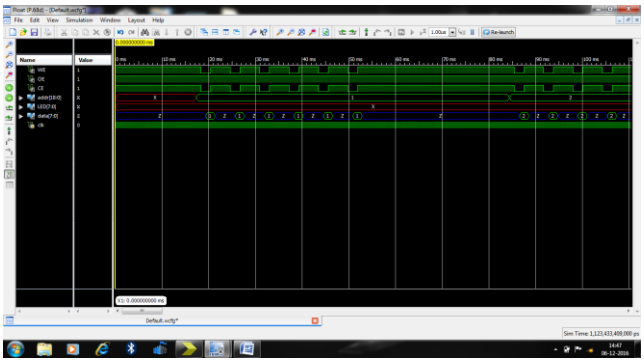


Fig. 10. Simulation result.

Fig.11. Power analysis from Xpower

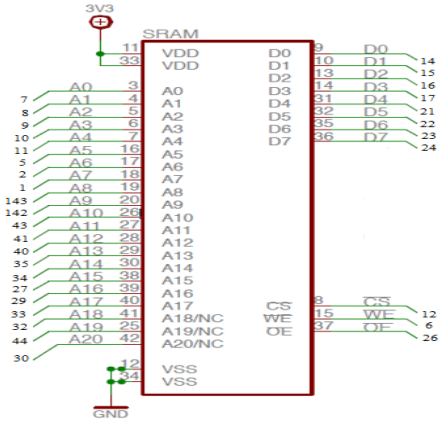


Fig. 12. SRAM pin diagram.

VII. HARDWARE PAPILIO DUO

Below table 1.shows the details of the device used for the implementation of the SRAM controller

Table 1. Device Details

Family	Spartan6
Part	Xc6slx9
Package	Tqg144
Temp Grade	C-Grade
Process	Typical
Speed Grade	-3

Below table 2. Shows the results of the implemented design

Table 2. Synthesis Report

On chip	Power (W)	Used	Available	Utilization (%)
Clock	0.001	4	-	-
Logic	0.000	113	5720	2
Signals	0.000	155	-	-
Ios	0.000	39	102	38
Leakage	0.014	-	-	-
Total	0.015	-	-	-

As shown in the table 1 are the details of the FPGA device used for the implementation of the assigned task.The table 2 summarises the details of the synthesis report. As per the table2 we can notice that the logic used to design is 113/5720 which as around 2% of the total resource available.Power consumed is also very negligible.

Fig. 13. SRAM IC.

The Papilio DUO board includes a 512KB ISSI IS61WV5128BLL or 2MB ISSI IS61WV20488BLL SRAM chip. SRAM is much, much easier to use with FPGA projects since there are no special timing requirements to follow. While we don't get as much SRAM memory space as we would for the same priced SDRAM or DDR memory chip, the trade off in ease of use more then makes up for it.

IX. CONCLUSION

SRAM has become the major component in VLSI design industry. For the projects in which power constrains are required like space exploration and satellites the SRAM cells can be used as they require less power consumption. There are many research going on SRAM controller in the VLSI industry by the scientists now a days.

REFERENCES

[1] Gupta, Vasudha, and Mohab Anis. "Statistical design of the 6T SRAM bit cell." IEEE Transactions on Circuits and Systems I: Regular Papers 57.1 (2010): 93-104.

[2] Chow, Paul, et al. "The design of an SRAM-based field-programmable gate array. I. Architecture." IEEE

Transactions on Very Large Scale Integration (VLSI) Systems 7.2 (1999): 191-197.

[3] 512KB ISSI IS61WV5128BLL SRAM-Datasheet

[4] http://www.cs.princeton.edu/courses /archive/spr06/cos116/FSM_Tutorial.pdf

[5] http://smithsonianchips.si.edu/ice/cd /MEMORY97/SEC08.PDF

[6] http://www.chips.ibm.com

[7] http://www.futureelectronics.com/en/memory/static-ram-sram.aspx

[8] www.issi.com

[9] Duo.Gadgetfactory.net

[10] www.wiki.com

[11] ftp://ftp.altera.com/up/pub/Altera_Material /12.1/University_Program_IP_ Cores/Memory/SRAM_Controller.pdf

[12] Lattice Mico Asynchronous SRAM Controller-device notes