University of Victoria

CENG 241

Digital Design I

Lab 7: RAM System

Instructor:
Dr. Amirali Baniasadi

Teaching Assistant:
Grace Hui

Yves Sénéchal V00213837
Tyler Stephen V00812021
A01 - B03

July 27, 2015

1 Introduction

The RAM controller reads and writes data to RAM modules. In this lab, the RAM controller was designed and implemented using D-FFs in a system similar to figure 1.

2 Discussion

The memory system built is shown in figure 1. The state diagram, included on page 3, determines the proper sequence to set the RAM controller into read or write mode. The RAM write operation cycle is summarized by the following:

- the RAM is set to accept data,
- the 4-bit counter generates an address for the data,
- the buffer is active and transmits its data to the RAM at the specified address,
- the register is in stand-by state.

The RAM read cycle is summarized by the following:

- the RAM is set to output data,
- the value of the 4-bit counter is the location of the read data,
- the buffer is set to high impedance mode which blocks all data to the RAM,
- the register is active and outputs the data received from the RAM.

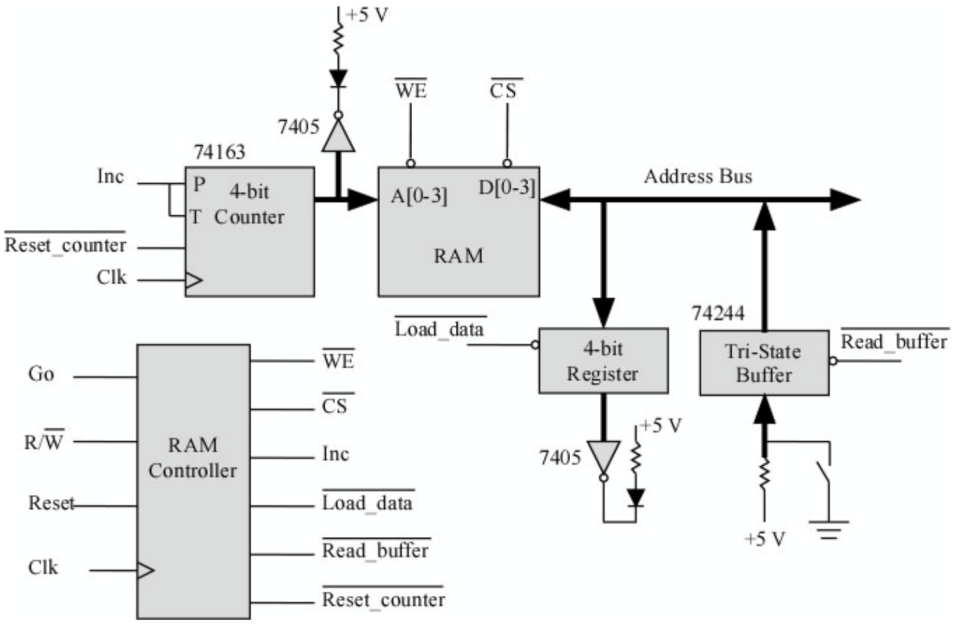


Figure 1: RAM controller and memory system

2.1 Timing considerations

The read and write timing diagrams, shown in figure 2 and figure 3 respectively, indicate that the moment in which the data is valid is offset from moment where the address is valid; the data

is valid after the address is valid. The access time and hold time for the read operation and write operation respectively can be attributed to the RAM's internal delay of 85 ns maximum. However, this does not pose a problem with a manual clock, but could become a nuisance should high-frequency clock be used.

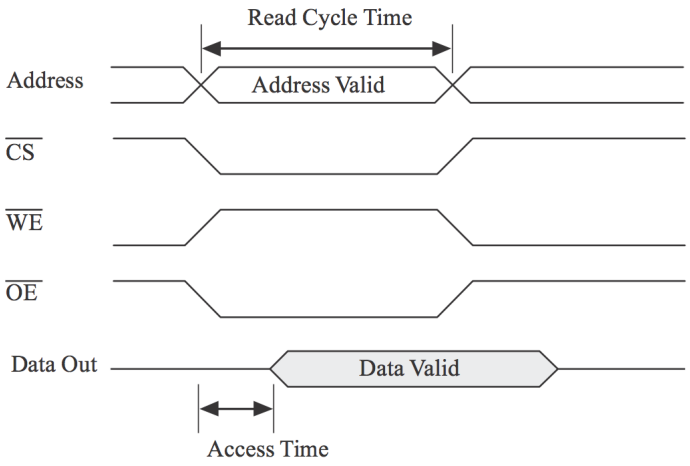


Figure 2: Timing diagram for the RAM read operation

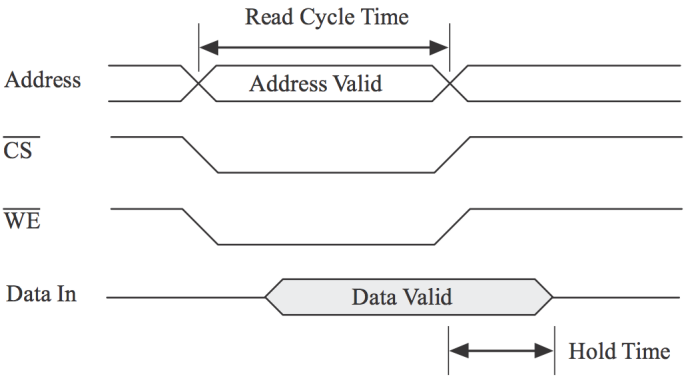
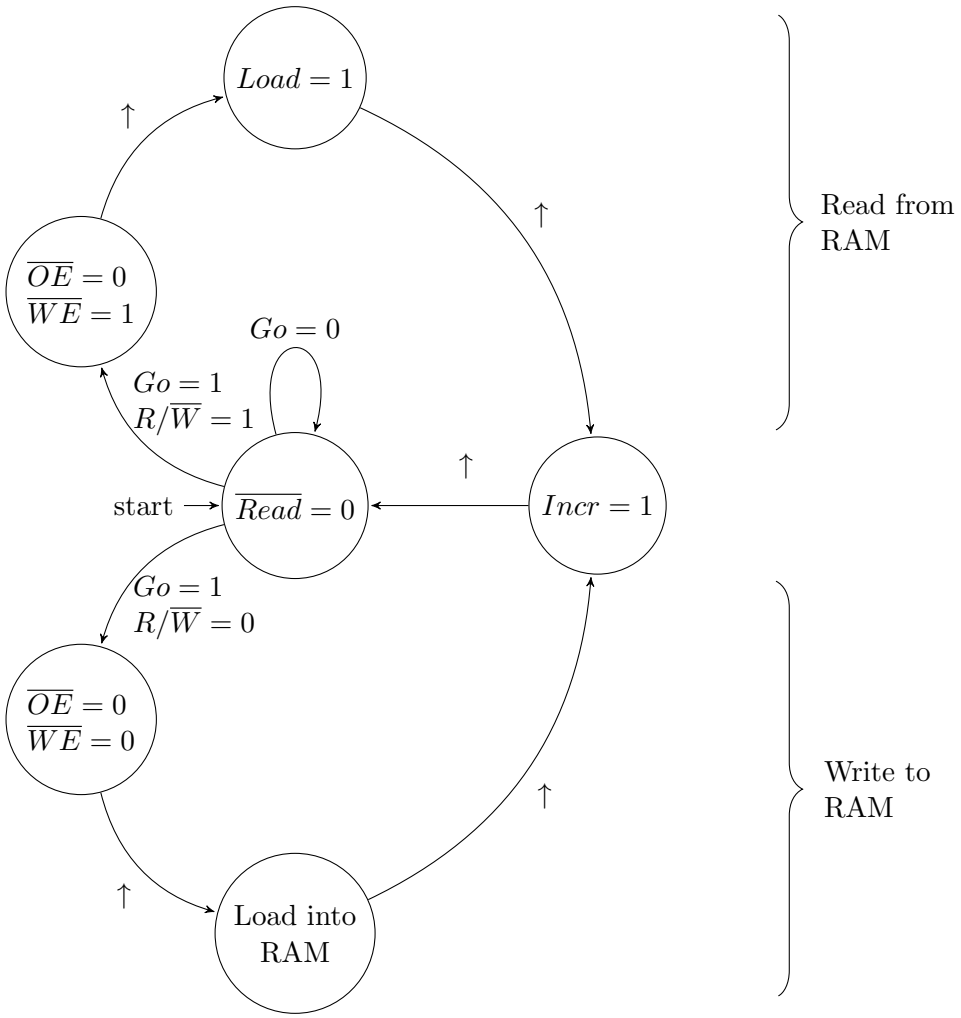


Figure 3: Timing diagram for the RAM write operation

3 RAM controller design process

3.1 State diagram

The state diagram of the RAM controller is shown below. The system was designed as a Moore machine with the transition tables displayed in figure 4.


$Load = 1$
$\overline{OE} = 0$
$\overline{WE} = 1$
$Go = 0$
$Go = 1$
$R/\overline{W} = 1$
start
$\overline{Read} = 0$
$Incr = 1$
$Go = 1$
$R/\overline{W} = 0$
$\overline{OE} = 0$
$\overline{WE} = 0$
Load into RAM
Read from RAM
Write to RAM

State	S_2	S_1	S_0
a	0	0	0
b	0	0	1
c	0	1	0
d	0	1	1
e	1	0	0
f	1	0	1
-	1	1	0
-	1	1	1

(a) State enumeration

S_2	S_1	S_0	Go	$R/\overline{W}$	S_2^+	S_1^+	S_0^+
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	1	0	0
0	0	0	1	1	0	0	1
0	0	1	0	0	0	1	0
0	0	1	0	1	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	1	0	1	0
0	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1
0	1	0	1	0	0	1	1
0	1	0	1	1	0	1	1
0	1	1	0	0	0	0	0
0	1	1	0	1	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	1	0	0	0
1	0	0	0	0	1	0	1
1	0	0	0	1	1	0	1
1	0	0	1	0	1	0	1
1	0	0	1	1	1	0	1
1	0	1	0	0	0	1	1
1	0	1	0	1	0	1	1
1	0	1	1	0	0	1	1
1	0	1	1	1	0	1	1
1	1	0	0	0	-	-	-
1	1	0	0	1	-	-	-
1	1	0	1	0	-	-	-
1	1	0	1	1	-	-	-
1	1	1	0	0	-	-	-
1	1	1	0	1	-	-	-
1	1	1	1	0	-	-	-
1	1	1	1	1	-	-	-

(b) Next state

Figure 4: Transition tables for the Moore machine

From here three karnaugh maps were used to resolve the next states S_2^+, S_1^+, and S_0^+. Their state equations are shown below.

$$S_2^+ = S_2 S_0' + S_1' S_0' \; Go \; R/\overline{W}$$
$$S_1^+ = S_1 \text{ XOR } S_0$$
$$S_0^+ = S_2 + S_1 S_0' + S_0' \; Go \; R/\overline{W}$$

S_2	S_1	S_0	$\overline{OE}$	$\overline{WE}$	$Load$	$\overline{Read}$	$Incr$
0	0	0	1	1	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	1	1	1	0
0	1	1	1	1	0	X	1
1	0	0	X	0	0	0	0
1	0	1	X	0	0	0	0
1	1	0	-	-	-	-	-
1	1	1	-	-	-	-	-

Table 1: State output

The state outputs are summarized in table 1. Five karnaugh maps were needed to resolve their state equations which are shown below.

$$\begin{aligned}
\overline{OE} &= S_1' S_0' + S_1 S_0 \\
\overline{WE} &= S_2' \\
Load &= S_1 S_0' \\
\overline{Read} &= S_1 + S_2' S_0 \\
Incr &= S_1 S_0
\end{aligned}$$

3.2 Controller logic

The schematic of the logic design for the RAM controller is included in figure 5. This design was implemented using three D-FFs plus some external logic.

4 Conclusion

A functioning 4-bit RAM controller was built using D-FFs and some external logic. The system had a 4-bit counter which determined the address of the data. Four LEDs displayed the present address, while, when in read mode, four other LEDs revealed the stored data at that address.

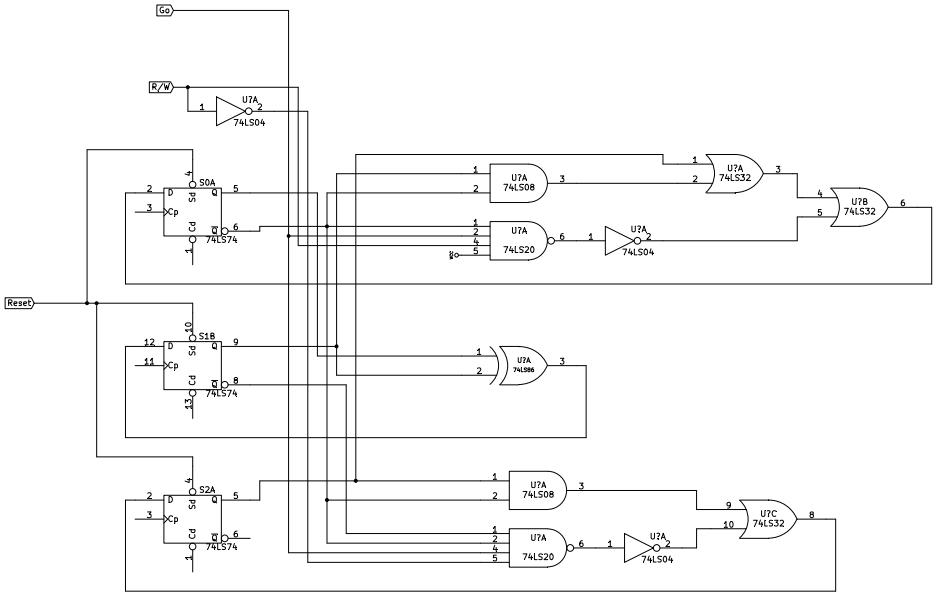


Figure 5: RAM controller logic design