ECE 385

Fall 2021

Experiment # 2

Data Storage

Xu Ke / Zhu Xiaohan

LA4/Thursday & 18:00-20:50 Huang Tianhao

I. Introduction

In this experiment, we learn about how to build structure for data-storage, which is very useful for future since storage is everywhere, processor always needs to store and shift data. More specifically, we design a 2-bit, 4-word shift-register storage unit with counters, comparators, multiplexers, and shift-registers. Furthermore, the function of the 2-bit, 4-word shift-register storage unit can store data into the corresponding address of the right shift register and read data at the given address of the register.

II. Operation of the memory circuit

1) Implementation of Addressing

Shown as the figure in the following part, here we use a 2-bit counter to track the corresponding data addresses. Since we want the counter and the shift-register update simultaneously, we let the counter and the shift register share the same clock to synchronize the circuit. In addition, we store the address of the rightmost register element of the shift register in the counter. In this case, if the data bits in the counter match the data bits in the SAR, the circuit will submit an instruction, either to perform read or write to the input switch and output register, while the FETCH or STORE switch will be turned on.

2) Implementation of write operation

The write operation will be performed in two steps: firstly the SAR switch is flipped to get the address, secondly the STORE switch will be flipped to write the data into memory. In our circuit, the comparator will determine whether the data bits in the SAR match the counter bits in each clock cycle or not. If they two same values, the SELECT signal will change value and select bits from the SBR in the 2-on-1 MUX, and thus data bits in the SBR can flow into the 4-bit shift register, otherwise, the 2-on-1 MUX will select the value from the rightmost element of the shift register and shifts in the loop.

3) Implementation of read operation

The read operation will also be performed in two steps: firstly the SAR switch for reading the address will be flipped, secondly the FETCH switch will be flipped to read the data from memory. In our circuit, counter will calculate the address for reading in each cycle. If the counter and the SAR bit have same value, the SELECT signal will change value and the MUX select a value from the right side of the shift register, and thus data from the shift register will flow into the SBR register, otherwise, the 3-to-1 MUX will select the value in the SBR for each cycle.

III. Written description and block diagram of memory circuit implementation

1) High-Level Description

To complete the memory circuit, we need a 2:1 MUX (74153) and two 4-bit shift registers (74194). The 2:1 MUX select signals from shift registers (shift right the old data) or SBR (load new data) according to STORE and COMP signals. The select bit logic will be discussed later. As said above, the output signal will be put into 2:1 MUX again for bit shifting and it will also be put into 3:1 MUX which might be chosen according to other signal to store in SBR. For the 3:1 MUX, we used a two-bit control

signal according to FETCH, COMP and LDSBR signals. It will decide to select data from SBR, Shift Register or DIN switches. Also, the SAR is used to store the address of the data. The total process is like:

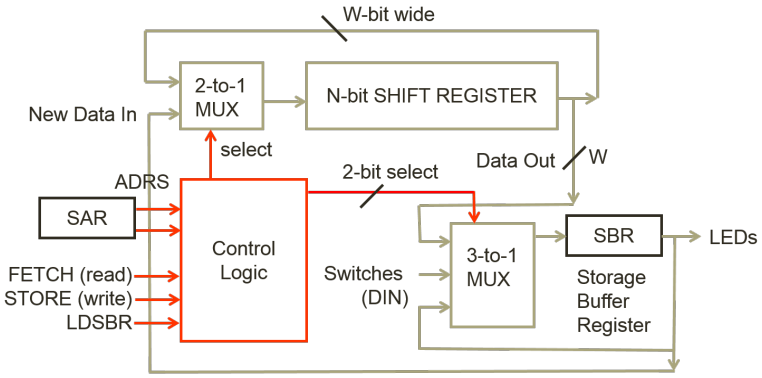
LDSBR: SBR <= DIN

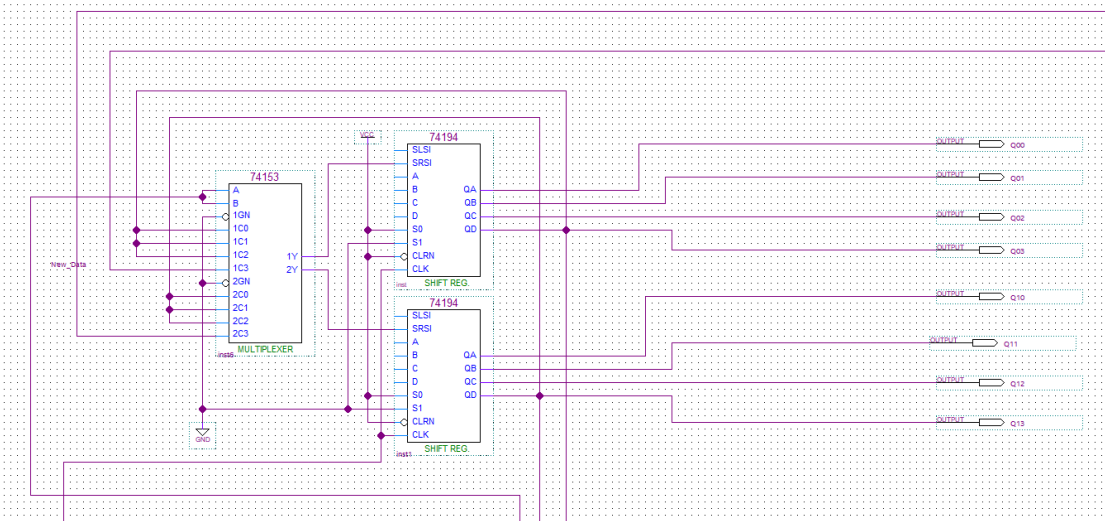
STORE: M[SAR] <= SBR

FETCH: SBR <= SAR

Always: LEDs <= SBR

2) High-Level Block Diagram

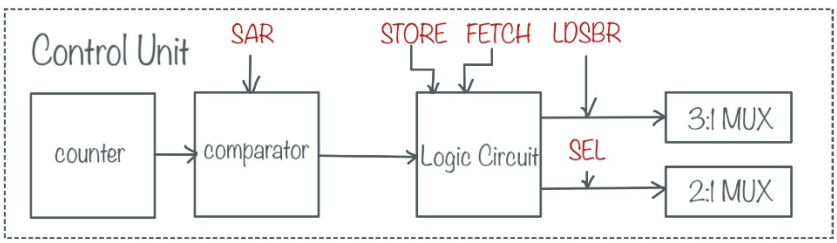


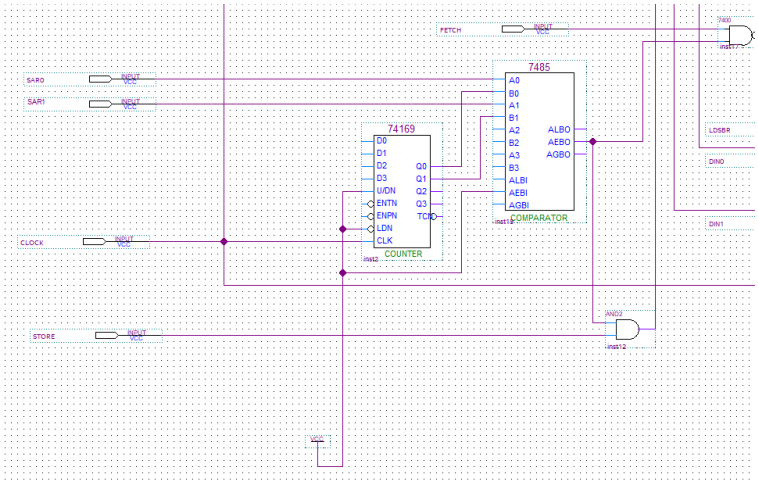


3) Control Unit

The Control unit is consisted of a counter (74169), a comparator (7485) and a logic circuit with input signals (CLOCK, FETCH, STORE, SAR). This control unit will process signal COMP and through a logic circuit to 3:1 MUX in order to select the output signals.

4) Block Diagram of Control Unit





IV. Design steps taken and detailed circuit schematic

1) K-maps or truth table

3:1 MUX

Port A: S0

Port B: S1

$S1 = \overline{(\overline{FETCH * COMP})} + \overline{LDSBR}$

$S2 = \overline{\overline{FETCH * COMP}}$

S1_S0 = 00 => Select from Shift Register

S1_S0 = 01 => Select from DIN Switch

S1_S0 = 11 => Select from SBR

LDSBR	COMP	FETCH	S1	S0	OUTPUT
0	0	0	1	1	SBR
0	0	1	1	1	SBR
0	1	0	1	1	SBR
0	1	1	0	0	SHIFT
1	0	0	0	1	DIN
1	0	1	0	1	DIN
1	1	0	0	1	DIN
1	1	1	0	0	SHIFT

2:1 MUX

SEL = STORE * COMP

SEL = 0 => Continue shifting right

SEL = 1 => Load new data from SBR

STOTE	COMP	SEL	OUTPUT
0	0	0	NEXT BIT
0	1	0	NEXT BIT
1	0	0	NEXT BIT
1	1	1	SBR

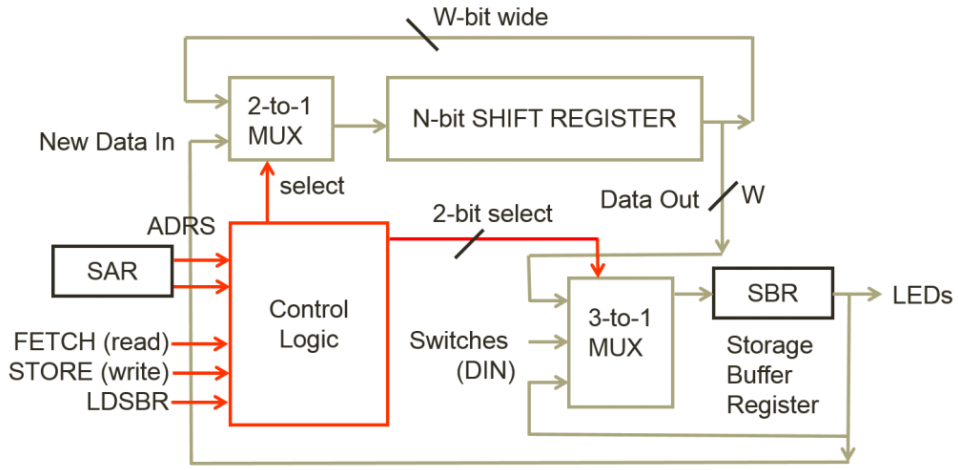
2) Written description of design considerations

For 3:1 MUX: We designed this select bit logic for convenience. If both COMP and FETCH are high, we load from shift register, otherwise, if LDSBR is low, we load from SBR, if LDSBR is high, we load from DIN switches.

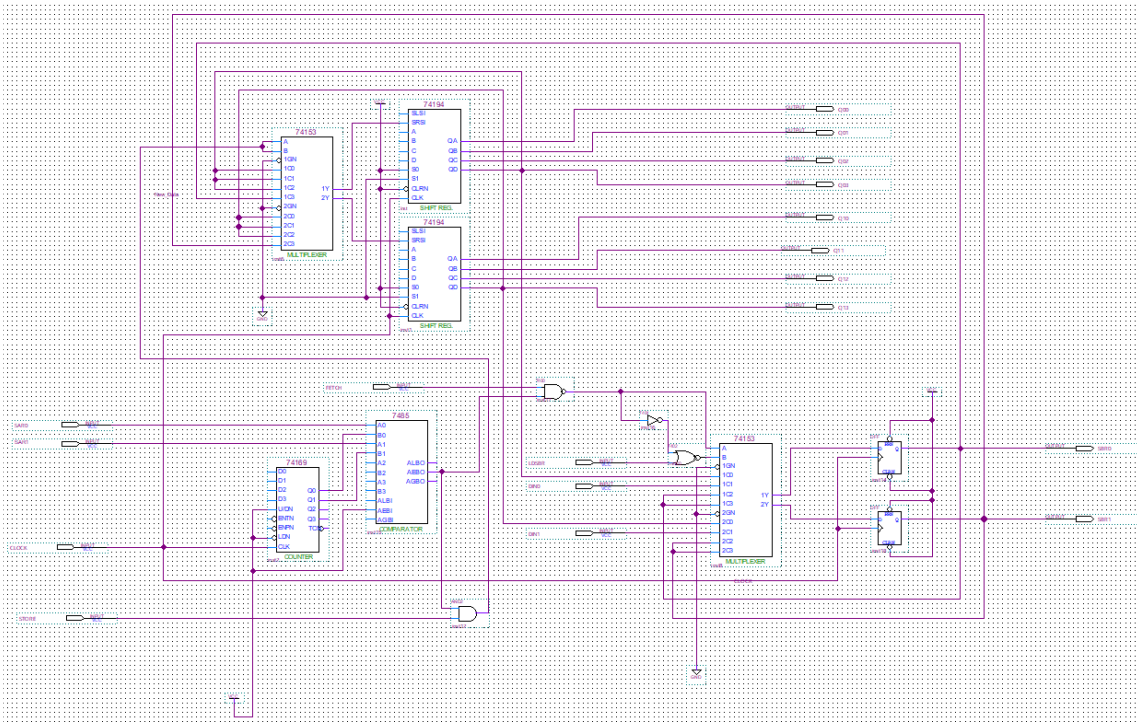
For 2:1 MUX: Only if both STORE and COMP signals are high, we will load from SBR, thus the logic design is simple as shown above.

3) Detailed Circuit Description

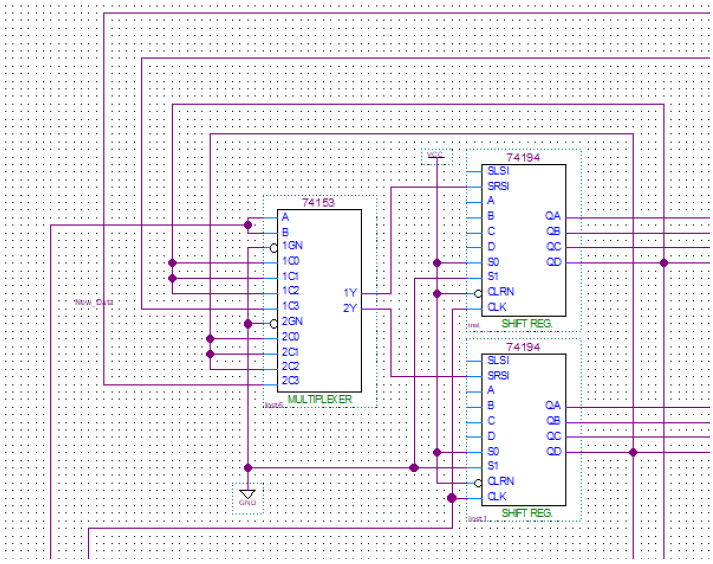
a) General Abstract Diagram



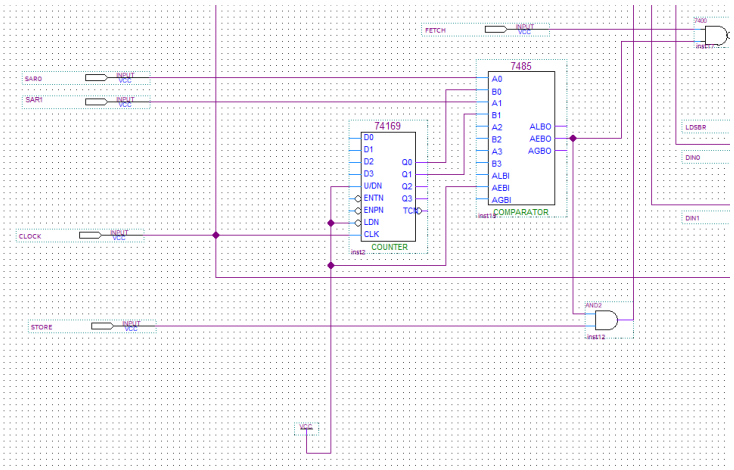
b) General Physical Diagram



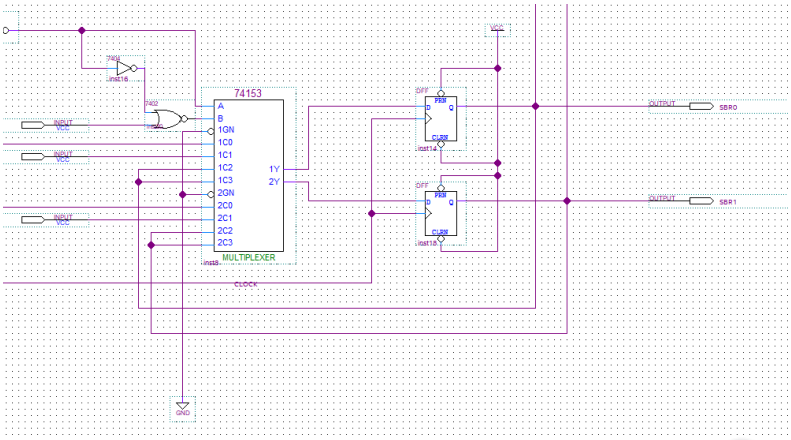
c) Memory Circuit + 2:1 MUX



d) Control Unit



e) SBR + 3:1 MUX



V. Description of all bugs encountered and corrective measures taken

We get wrong with the selection signal port A and port B of 74153, that is, port A should be S0 and port B should be S1, but we got it backwards. So that our logic circuit design is not correct. We correct this fault by analyze the output signals with input

signals and redesign the logic circuit.

VI. Answers to Post-Lab Question

1) Question 1:What are the performance implications of your shift register memory as compared to a standard SRAM of the same size?

Answer: Compared to a standard SRAM of the same size, our shift register memory (DRAM) is different in several aspects and thus lead to performance implications. At first, our circuit as a DRAM can periodically shift or update (i.e. write) the data, while a standard SRAM refers to Static Random-Access Memory, which is in static state once set up and cannot be easily updated. Secondly, our circuit uses less elements, has more simple structure and takes less volume which SRAM cannot enjoy these advantages. Last but not least, SRAM is much faster than our circuit since our circuit has to wait for the load-in data address.

2) Question 2:What are the implications of the different counters and shift register chips, what was your reasoning in choosing the parts you did?

Answer: We choose to use counter 74169 and shift register 74194. The reason for choosing them is because we want to make sure that in our circuit all counters and shift register chips we use are synchronous thus the data bits can be shifted at the same time and all actions can happen on the rising edge of the shared clock to work correctly. Otherwise, all things will be in a mess and data-corruption may take place.

VII. Conclusion

From this experiment, we learn about how to build structure for data-storage, more specifically, we design a 2-bit, 4-word shift-register storage unit with counters, comparators, multiplexers, and shift-registers. This experiment is very useful for future since storage is everywhere, processor always needs to store and shift data. Furthermore, this experiment helps us better understand about how to implement a control logic correctly and put what we have learned into practice and allow us to choose elements and build circuits in our own way.