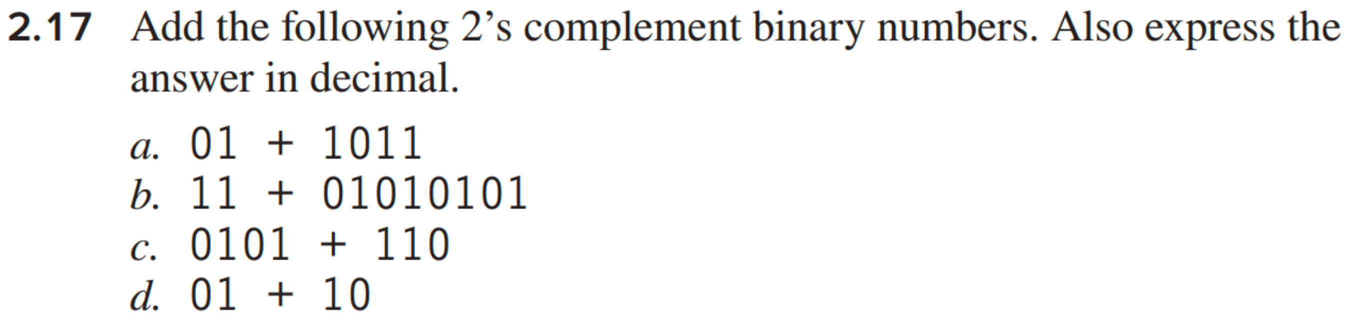
HW 1



2.17 Add the following 2's complement binary numbers. Also express the answer in decimal.

a. 01 + 1011
b. 11 + 01010101
c. 0101 + 110
d. 01 + 10

> 符号位扩展
> 结果以补码表示

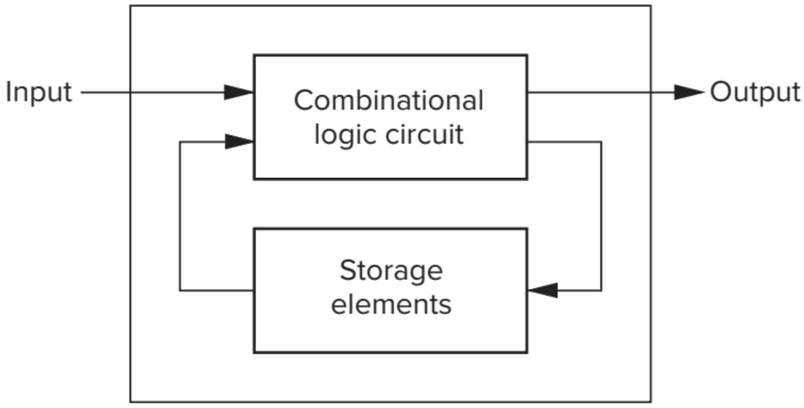
CH2

Finite State Machine(FSM)

- **state**

  > The state of a system is a snapshot of all the relevant elements of the system at the moment the snapshot is taken.
  >
  > e.g. **soft drink machine**/basketball game/ tic-tac-toe(3^9 states)

Sequential Logic Circuits



Sequential logic circuit block diagram.

> That is, these structures base their decisions not only on the input values now present, but also (and this is very important) on what has happened before. These structures are usually called **sequential logic circuits**.

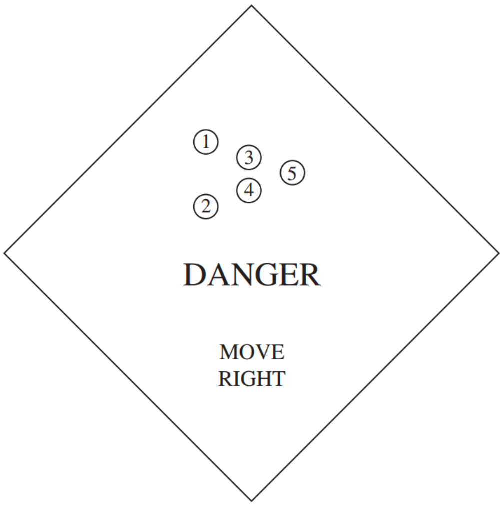
Synchronous Finite State Machine

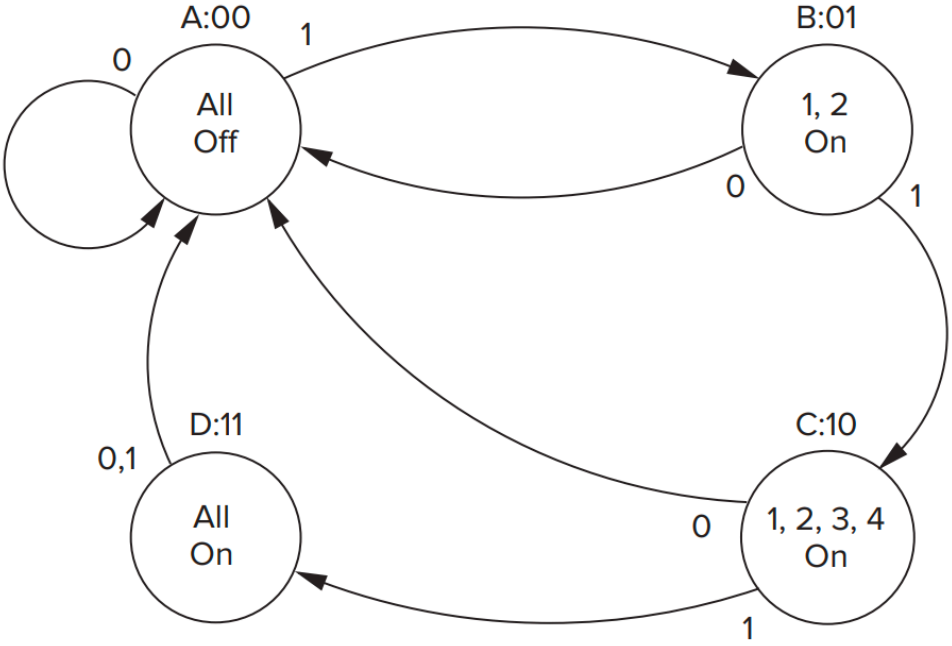
- **Synchronous VS Asynchronous**

  > Synchronous: 固定的时间改变一次状态 $clock\ cycle$

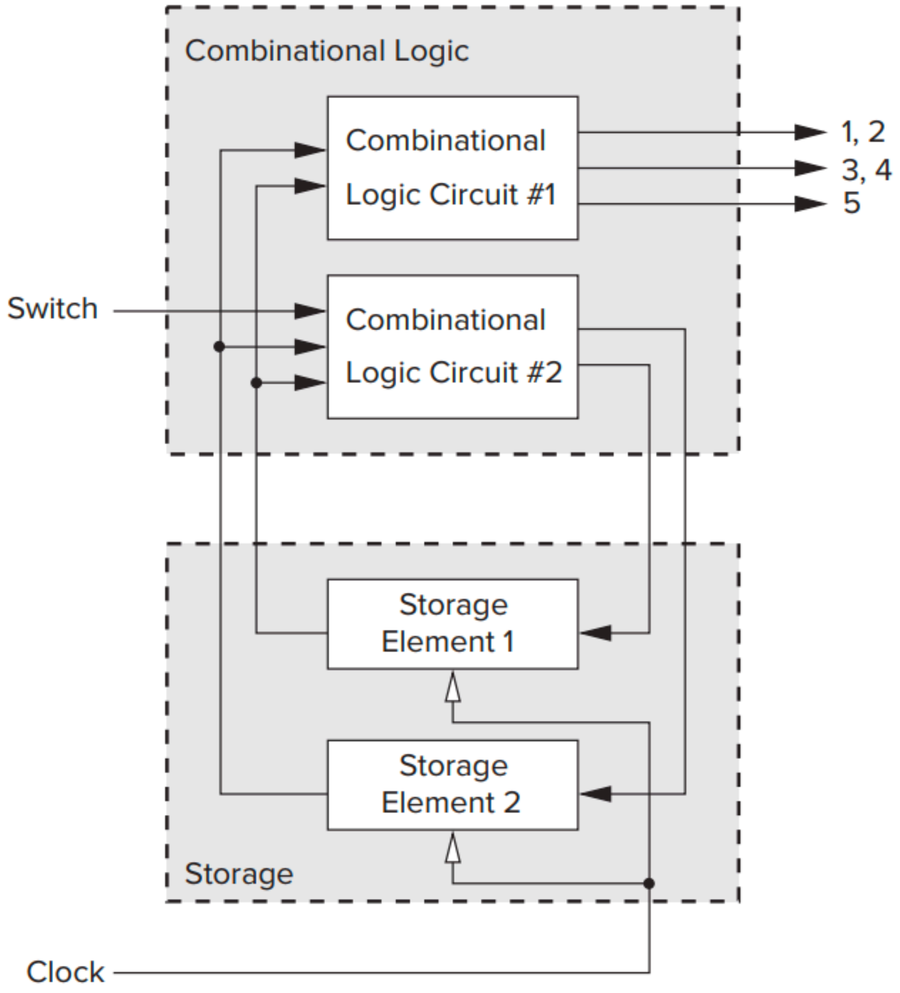
- Danger Sign

- State diagram





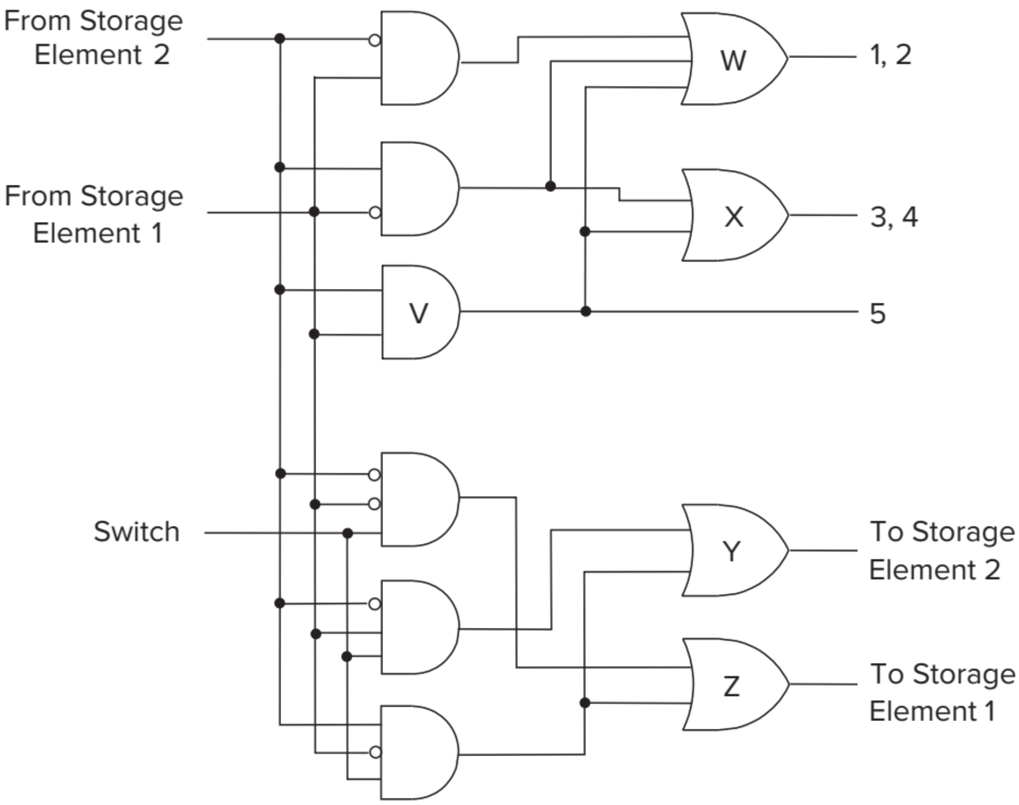
- Logic Circuit



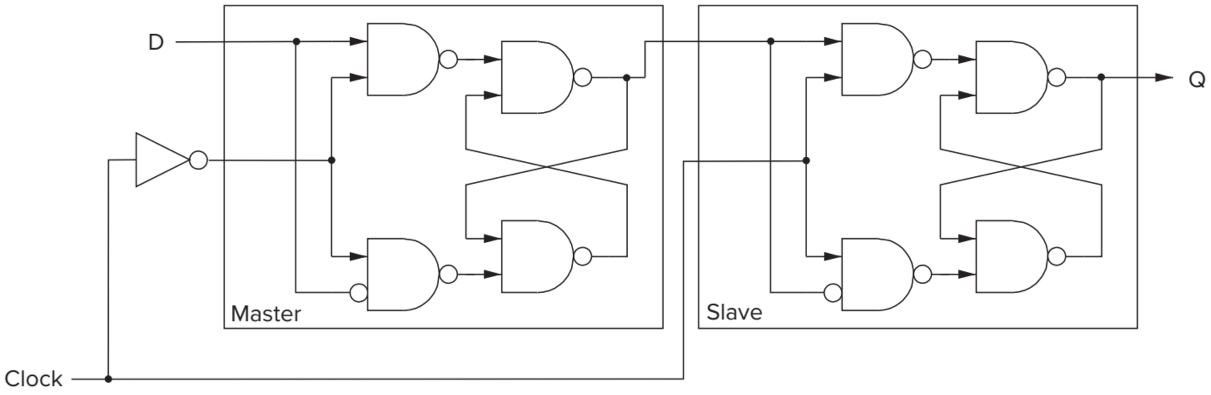
- Combinational Logic Circuit

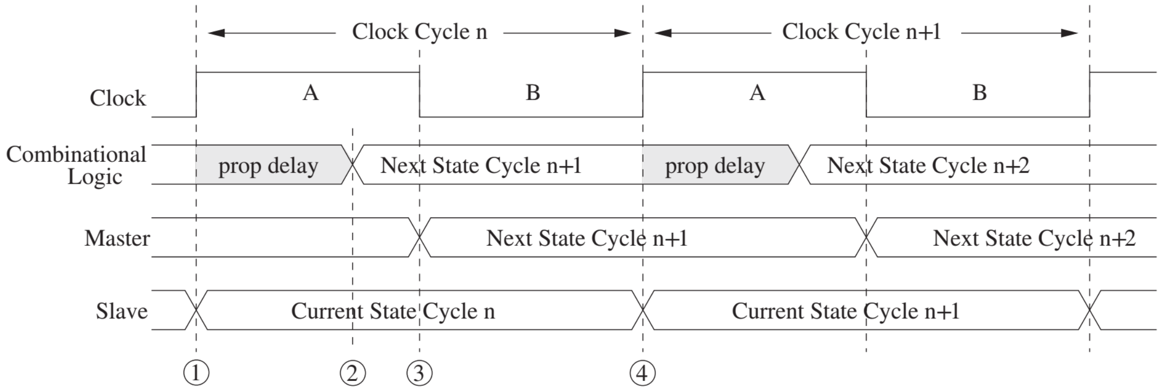
1. PLA的设计方法：先画真值表，然后设计。
2. 可以省略$decoder$中没有用的输出。
3. 可以优化。

$Switch$	S_1	S_0	S_1'	S_0'
0	X	X	0	0
1	0	0	0	**1**
1	0	1	**1**	0
1	1	0	**1**	**1**
1	1	1	0	0

> 
>
>
> **3.32 Combinational logic circuits 1 and 2.**

- master/slave flip-flop

> - A Problem with Gated Latches as Storage Elements
>
> 问题在于$D-Latch$是$Transparent$的，面对组合逻辑它会立刻反应出来，导致一个周期更新多次$state.$
>
> - **master/slave flip-flop**
>
> 能够保持同一个$state$一个周期（$Master$类似于$Slave$的$buffer$）
>
> 下降沿把$state_{i+1}$写入$Master$但是$Slave$并没有改变，仍然是$state_i$
>
> 上升沿写入$Slave$但是$Master$无法改变，因此它的状态$state_{i+1}$也无法在这个上升沿改变
>
> 
>
>
> **Figure 3.33 A master/slave flip-flop.**

> 
>
>
> **Figure 3.34** Timing diagram for a master/slave flip-flop.

CH4 The von Neumann Model

Basic Component

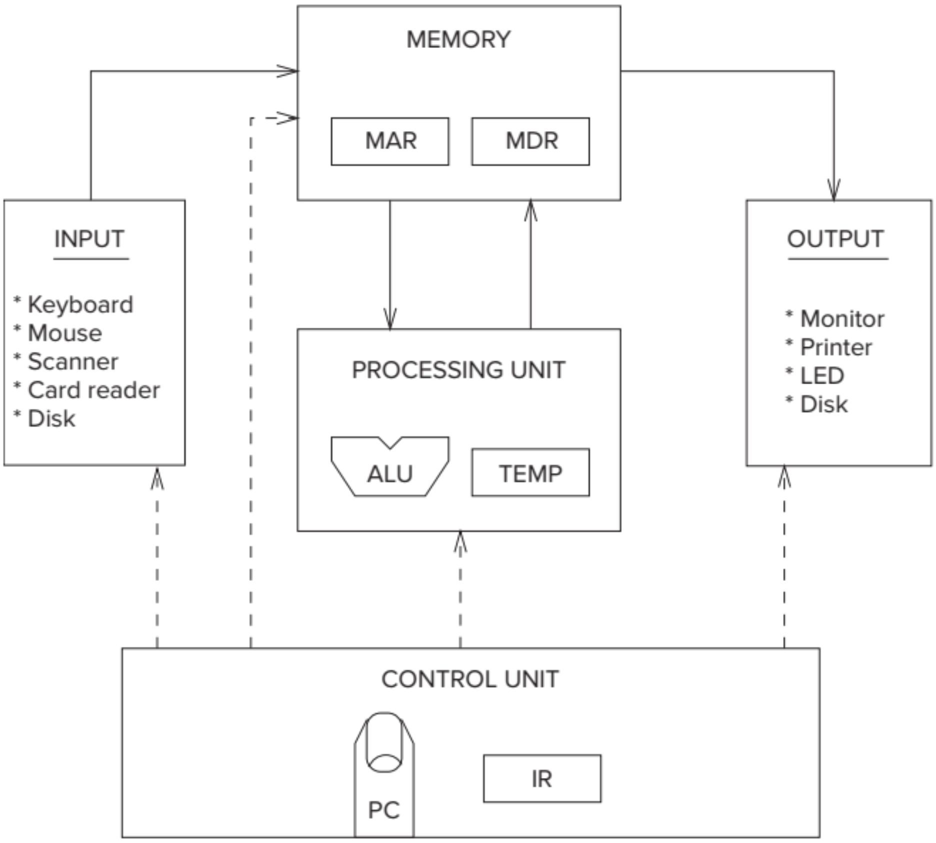


Figure 4.1 The von Neumann model, overall block diagram.

- Processing Unit/Computation Unit

  > - 需要有$TEMP$储存中间结果，我们这里使用的是$register$寄存器来暂存值来加快运算速度
  > - $LC-3$ has 8 registers，可以用3位二进制标识，每个寄存器可以存16位的值（由16个$flip-flop$构成）

- Memory

  > - $LC-3$有2^{16}个地址，每个地址能存16 bits.

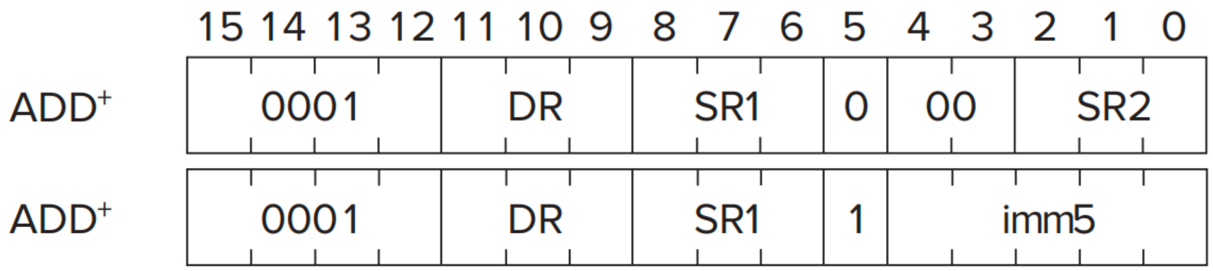
> - 指令和数据存在一起

- Control Unit

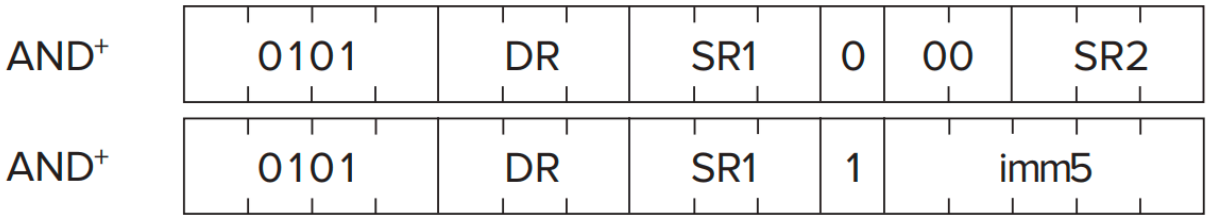
> - PC : Program Counter
>
> > 实际上代表指令地址
>
> - IR : Instruction Register

Instruction（Appendix A）

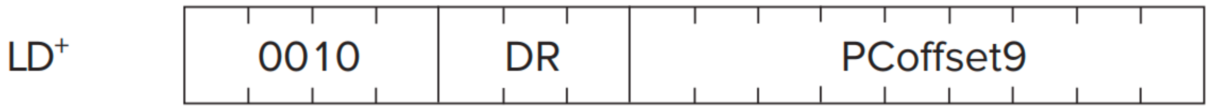
- ADD

> 
>
> | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
> |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
> | ADD+ | 0001 | | | | DR | | | SR1 | | | 0 | 00 | | SR2 | | |
> | ADD+ | 0001 | | | | DR | | | SR1 | | | 1 | imm5 | | | | |

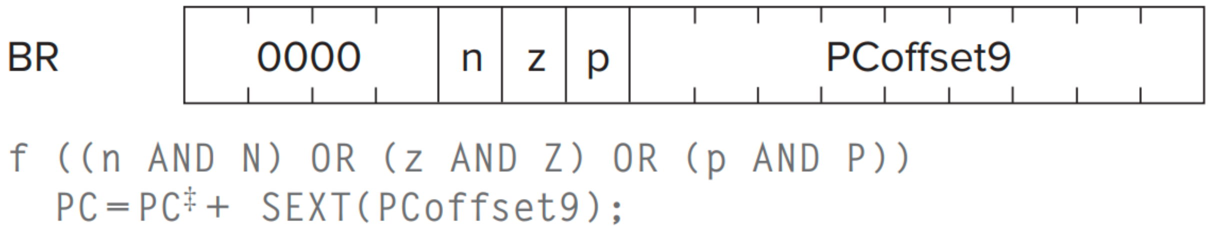
- AND

> 
>
> | | | | | | | | | | | | | | | | | |
> |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
> | AND+ | 0101 | | | | DR | | | SR1 | | | 0 | 00 | | SR2 | | |
> | AND+ | 0101 | | | | DR | | | SR1 | | | 1 | imm5 | | | | |

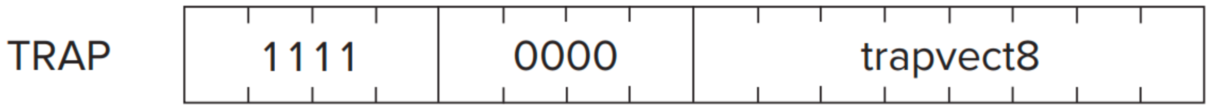
- LD

> 
>
> | | | | | | | | | | | | | | | | | |
> |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
> | LD+ | 0010 | | | | DR | | | PCoffset9 | | | | | | | | |
>
> 在**已经自增**的PC上加SEXT(PCoffset)

- BR

> 
>
> | | | | | | | | | | | | | | | | | |
> |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
> | BR | 0000 | | | | n | z | p | PCoffset9 | | | | | | | | |
>
> ```
> if ((n AND N) OR (z AND Z) OR (p AND P))
>    PC=PC‡+ SEXT(PCoffset9);
> ```

- TRAP

> 
>
> | | | | | | | | | | | | | | | | | |
> |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
> | TRAP | 1111 | | | | 0000 | | | | trapvect8 | | | | | | | |

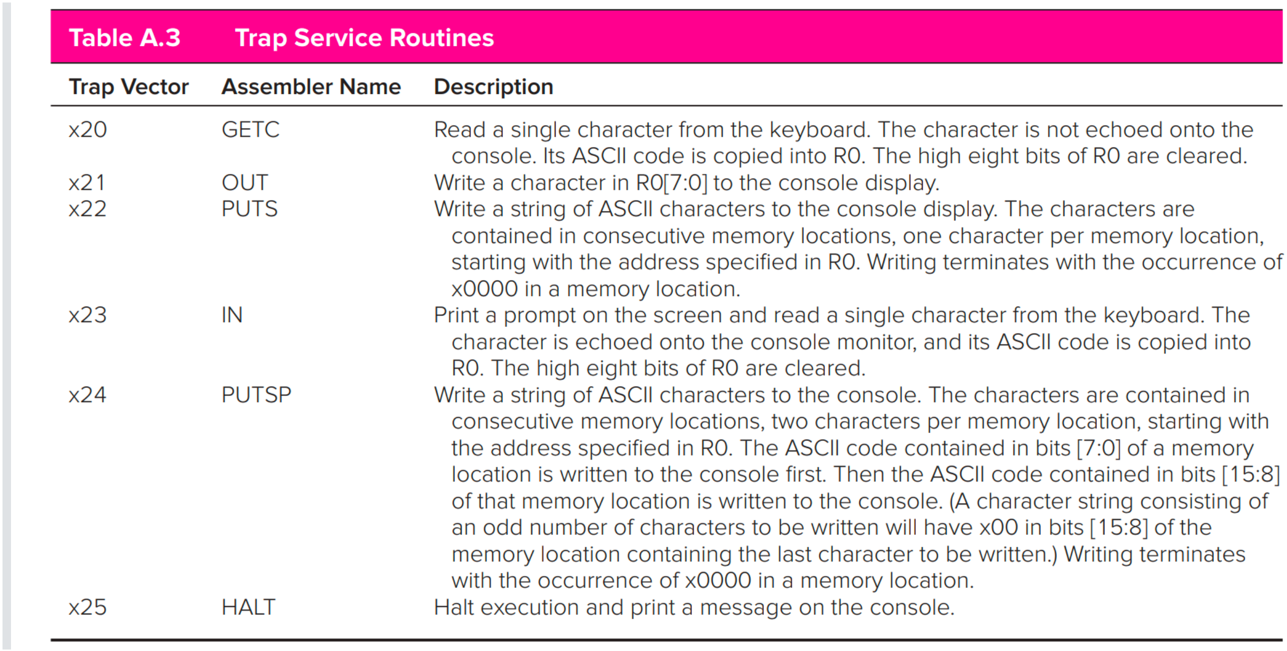
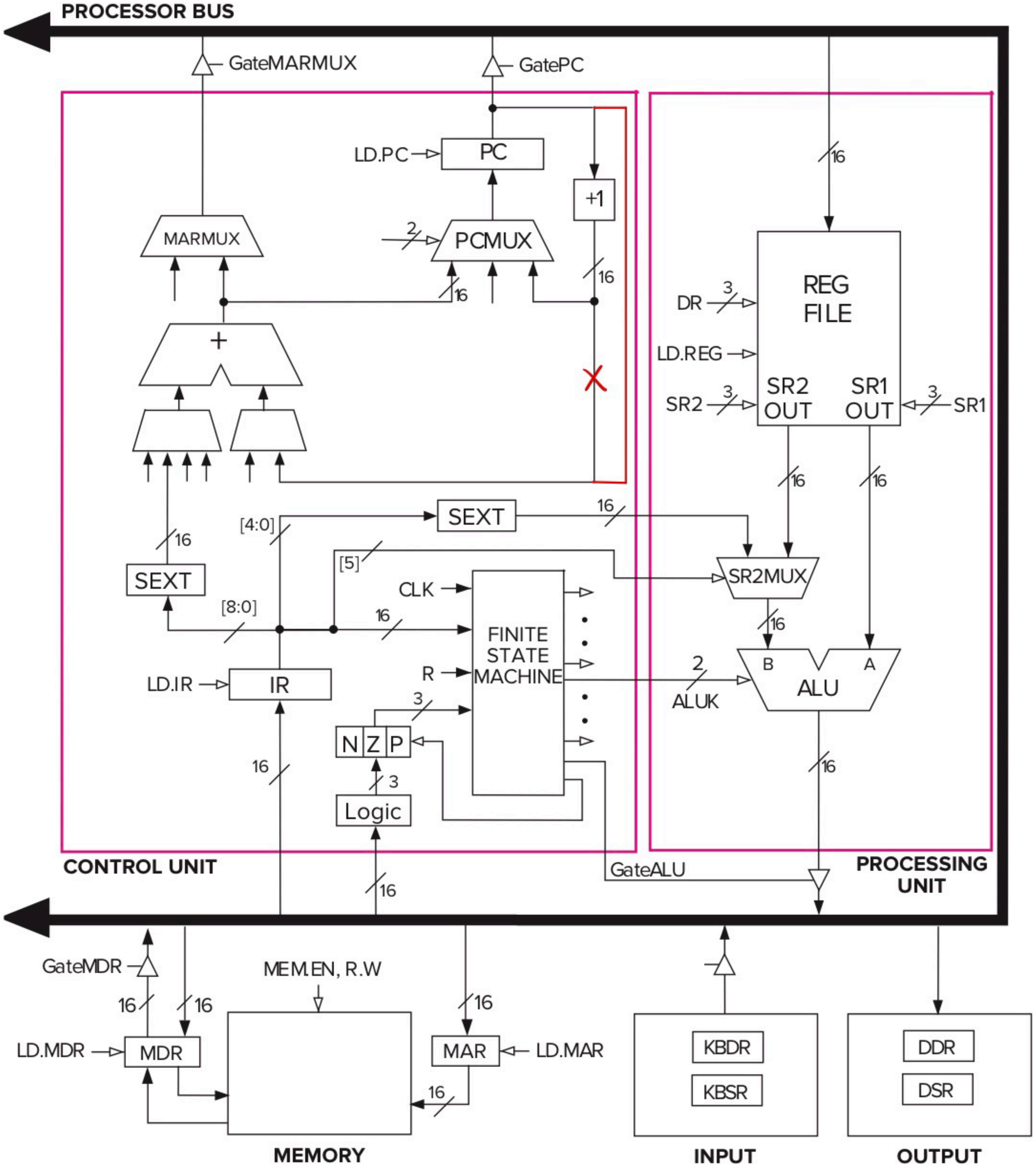


Table A.3 Trap Service Routines

Trap Vector	Assembler Name	Description
x20	GETC	Read a single character from the keyboard. The character is not echoed onto the console. Its ASCII code is copied into R0. The high eight bits of R0 are cleared.
x21	OUT	Write a character in R0[7:0] to the console display.
x22	PUTS	Write a string of ASCII characters to the console display. The characters are contained in consecutive memory locations, one character per memory location, starting with the address specified in R0. Writing terminates with the occurrence of x0000 in a memory location.
x23	IN	Print a prompt on the screen and read a single character from the keyboard. The character is echoed onto the console monitor, and its ASCII code is copied into R0. The high eight bits of R0 are cleared.
x24	PUTSP	Write a string of ASCII characters to the console. The characters are contained in consecutive memory locations, two characters per memory location, starting with the address specified in R0. The ASCII code contained in bits [7:0] of a memory location is written to the console first. Then the ASCII code contained in bits [15:8] of that memory location is written to the console. (A character string consisting of an odd number of characters to be written will have x00 in bits [15:8] of the memory location containing the last character to be written.) Writing terminates with the occurrence of x0000 in a memory location.
x25	HALT	Halt execution and print a message on the console.

Example: LC-3



- 冯诺依曼架构
- The Instruction Cycle **(NOT the Clock Cycle!)**

> （以ADD，LD，BR为例）
>
> - FETCH
>
>   > 1个时钟周期把PC存入MAR（通过BUS），同时PC=PC+1
>   > 5个周期（可能更多）把数据放入MDR
>   > 1个时钟周期把MDR的数据送回IR（通过BUS）

> - DECODE
> - EVALUATE ADDRESS
> - FETCH OPERANDS
> - EXECUTE
> - STORE RESULT

- Finite State Machine

  > 状态机决定了各个控制信号的值。

- Tip

  > 为什么Reg File要使用flip flop