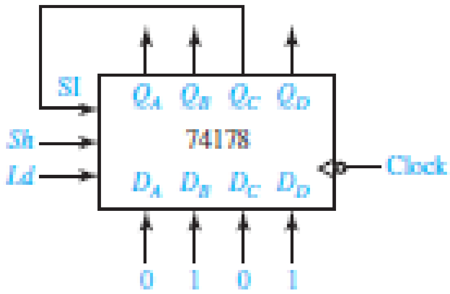
Spring 2015

BSM 206 Computer Organization

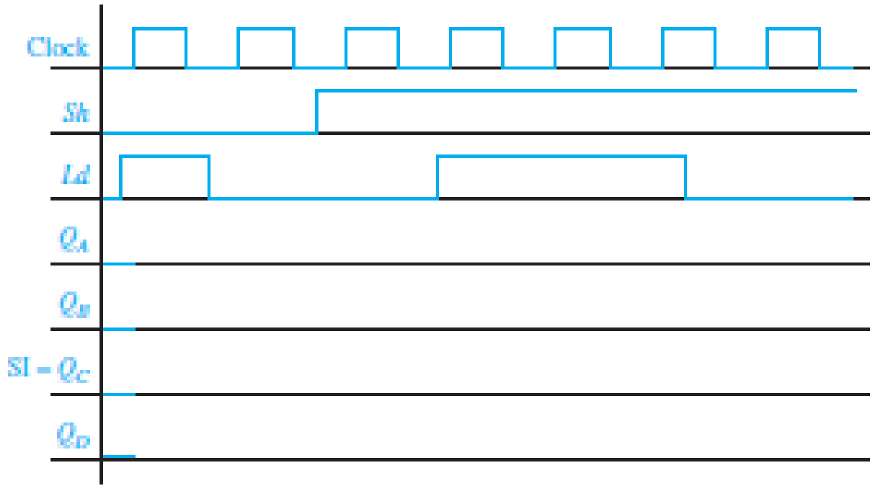
Homework #1 Solutions

Questions:

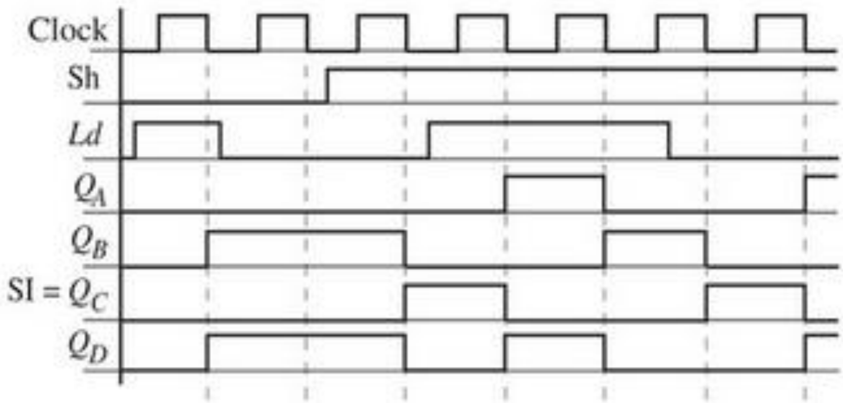
1) [15points] [Registers] A 74178 shift register is described by the given table. All state changes occur on the 1-0 transition of the clock. The shift register is connected as shown. Complete the timing diagram.



Sh	Ld	Q_A^+	Q_B^+	Q_C^+	Q_D^+
0	0	Q_A	Q_B	Q_C	Q_D
0	1	D_A	D_B	D_C	D_D
1	X	SI	Q_A	Q_B	Q_C



The answer is shown below.



2) [25points] [Computer Abstractions and Technology] Consider two different implementations, M1 and M2, of the same instruction set. There are three classes of instructions (A, B, and C) in the instruction set. M1 has a clock rate of 80 MHz and M2 has a clock rate of 100 MHz. There is a specific program can be written in two different sequences. The average number of cycles for each instruction class and their frequencies (for each sequence) are as follows:

Sequence	Instruction Class	Machine M1 – Cycles /Instruction Class	Machine M2 – Cycles /Instruction Class	Frequency
Sequence 1	A	1	2	60%
	B	2	3	30%
	C	4	4	10%
Sequence 2	A	1	2	20%
	B	2	3	40%
	C	4	4	40%

a) Calculate the average CPI for each machine and for each sequence; M1-Sequnce 1, M1-Sequence 2, M2-Sequence 1, and M2-Sequence-2.

For Machine M1 and Sequence 1:

Clocks per Instruction = (60/100) x 1 + (30/100) x 2 + (10/100) x 4
= 1.6

For Machine M2 and Sequence 1:

Clocks per Instruction = (60/100) x 2 + (30/100) x 3 + (10/100) x 4
=2.5

For Machine M1 and Sequence 2:

Clocks per Instruction = (20/100) x 1 + (40/100) x 2 + (40/100) x 4
= 2.6

For Machine M2 and Sequence 2:

Clocks per Instruction = (20/100) x 2 + (40/100) x 3 + (40/100) x 4
=3.2

b) If we assume that the execution time is equal to CPU time, compare these two machines' performances for each sequence. Which one is faster and how much?

Let I denote the instruction count.

For Sequence 1:

CPU_{time} for M1 = 1.6I / 80MHz = 0.02×10^{-6} I

CPU_{time} for M2 = 2.5I / 100MHz = 0.025×10^{-6} I

In this case, M1 is faster than M2.

By how much: $Performance_{M1}$ / $Performance_{M2}$ = Execution $time_{M2}$ / Execution $time_{M1}$

= CPU_{time} for M2 / CPU_{time} for M1

= 0.025 / 0.02

= 1.25

For Sequence 2:

CPU_{time} for M1 = 2.6I / 80MHz = 0.0325×10^{-6} I

CPU_{time} for M2 = 3.2I / 100MHz = 0.032×10^{-6} I

In this case, M2 is faster than M1.

By how much: $Performance_{M2}$ / $Performance_{M1}$ = Execution $time_{M1}$ / Execution $time_{M2}$

= CPU_{time} for M1 / CPU_{time} for M2
= 0.0325 / 0.032
= 1.015

c) Based on this information, calculate the average MIPS ratings for each machine and for each sequence, M1 and M2.

For machine M1 and Sequence 1:

Average MIPS rating = I / (CPU_{time} x 10^6) = I / [(1.6I / 80.10^6) x 10^6]= 50

For machine M1 and Sequence 2:

Average MIPS rating = I / (CPU_{time} x 10^6) = I / [(2.6I / 80.10^6) x 10^6]= 30.77

For machine M2 and Sequence 1:

Average MIPS rating = I / (CPU_{time} x 10^6) = I / [(2.5I / 100.10^6) x 10^6]= 40

For machine M2 and Sequence 2:

Average MIPS rating = I / (CPU_{time} x 10^6) = I / [(3.2I / 100.10^6) x 10^6]= 31.25,

d) Which machine has a smaller MIPS rating for sequence 1? Which individual instruction class CPI do you need to change, and by how much, to have this machine have the same or better performance as the machine with the higher MIPS rating (you can only change the CPI for one of the instruction classes on the slower machine)?

Machine M2 has a smaller MIPS rating. If we change the CPI of instruction class A for Machine M2 to 1, we can have a better MIPS rating than M1 as follows:

Clocks per Instruction = (60/100) x 1 + (30/100) x 3 + (10/100) x 4
= 1.9

Average MIPS rating = Clock Rate/(CPI * 10^6)
= (100 * 10^6) / (1.9*10^6)
= 52.6

e) Repeat (d) for sequence 2.

Machine M1 has a smaller MIPS rating. If we change the CPI of instruction class B for Machine M1 to 1, we can have a better MIPS rating than M1 as follows:

Clocks per Instruction = (20/100) x 1 + (40/100) x 1 + (40/100) x 4
= 2.2

Average MIPS rating = Clock Rate/(CPI * 10^6)
= (80 * 10^6) / (2.2*10^6)
= 36.6

3) [20points] [Instructions: Language of Computers] Consider the following sequence of code:
z = (x + y) - (z - q);
Assume that x, y, z, q are stored in registers $s2, $s3, $s4, $s5.

a) Write the code into MIPS assembler. Please use the box provided below.
b) Complete the machine code and memory addresses given below for these instructions.

```
add $t0, $s2, $s3

sub $t1, $s4, $s5

sub $s4. $t0. $t1
```

Memory	op	rs	rt	rd	shamt	funct
40004	0	17	18	8	0	32
40008	0	19	20	9	0	34
40012	0	8	9	19	0	34

4) [40points] [Computer Abstractions and Technology] Consider the following C function:

```
int DownThreeUntilZero(int m)
{
  if (m < 3) return 0;
  else return m + DownThreeUntilZero(m - 3);
}
```

a) The argument m is stored in $a0. Provide the MIPS code. Please put a comments for each code.

```
dtuz :  addi $sp, $sp, -8     # adjust stack for 2 items
        sw   $ra, 4($sp)      # save return address
        sw   $a0, 0($sp)      # save argument
        slti $t0, $a0, 3      #  test for m < 3
        beq $t0, $zero, L1    # go L1 if m >= 3
        addi $v0, $zero, 0    # if m < 3, result is 0
        addi $sp, $sp, 8      # pop 2 items from stack
        jr   $ra              # and return
L1:     addi $a0, $a0, -3     #  else set the argument to m-3
        jal  dtuz             # recursive call
        lw   $a0, 0($sp)      # restore original m
        lw   $ra, 4($sp)      # restore return address
        addi $sp, $sp, 8      # pop 2 items from stack
        add  $v0, $a0, $v0    # add to get the result
        jr   $ra              # and return
```

b) For this MIPS code, provide the machine code as shown below. Please add as much rows as needed for the program. The memory addresses of the first two instructions are given. Please write memory addresses for each instructions. The first two instructions are given in R-format, please reshape the row for different formats by merging cells.

Memory	op	rs	rt	rd	shamt	funct
				constant or address		
100016	8	29	29	-8		
100020	43	29	31	4		
100024	43	29	4	0		
100028	10	4	8	3		
100032	4	8	0	3		
100036	8	2	0	0		
100040	8	29	29	8		
100044	0	31	0	0	0	8
100048	8	4	4	-3		
100052	3	25004				
100056	35	29	4	0		
100060	35	29	31	4		
100064	8	29	29	8		
100068	0	4	2	2	0	32
100072	0	31	0	0	0	8