University of New Brunswick

Computer Science

CS3853: Computer Architecture and Organization

Instructor: Joannah Nanjekye, jnanjeky@unb.ca

Due Date: July 22, 2024 – 11:59 PM

ASSIGNMENT 1

Submission instructions:

- Submit a pdf file to the Desire2Learn dropbox

Problem 1. Given the instruction set of the IAS computer:

- Write an assembly language program that loads the integers 6,1,8,2, and 3 from location 300 to 304. The program should sort the contents in the memory locations such that memory location 300 has the lowest value while 304 has the highest value. Write your assembly language programming code in a 3-column format: Address, Opcode (Symbolic), and Operand.

  Solution: 5 points

  0 JUMP(305, 0:19)
  300 6
  301 1
  302 8
  303 2
  304 3

305	LOAD	MQ,M(300)
	LOAD	M(301)
306	STOR	M(300)
	LOAD	M(303)
307	STOR	M(301)
	LOAD	MQ
308	STOR	M(303)
	LOAD	MQ, M(302)
309	LOAD	M(304)
	STOR	M(302)
310	LOAD	MQ
	STOR	M(304)
311	JUMP	M(0,0:19)

- Assume that each instruction takes 3 clock cycles to execute (1 fetch, 1 decode, and 1 execute) and a 3MHz clock rate. Calculate the execution time.

  Solution: 3 points

$$CPU\ Time = \frac{CPI \times InstructionCount}{Clock\ Rate}$$
$$= \frac{14 \times 3}{3 \times 10^6}$$

$= 1.4 \times 10^-5s$

Problem 2. A compiler developer is trying to compare the designs of three machines with their respective CPIs for several instruction categories. All machines have the same instruction set.

Instruction Category	CPI_A	CPI_B	CPI_C	Instruction Count
Load/Store	12	11	9	1
Subtract	8	7	5.6	2
Jump	5	4	7	3
Branch	9	7	12	4
Shift	6	5	9	5
Other	22	12	17	3

(a) Calculate the average CPI for each machine.

Solution: 4 points

$CPI_A = \frac{(12\times1)+(8\times2)+(5\times3)+(9\times4)+(6\times5)+(22\times3)}{18} = 9.72$

$CPI_B = \frac{(11\times1)+(7\times2)+(4\times3)+(7\times4)+(5\times5)+(12\times3)}{18} = 7$

$CPI_c = (9\frac{\times1)+(5.6\times2)+(7\times3)+(12\times4)+(9\times5)+(17\times3)}{18} = 10.29$

(b) What is the clock rate for each machine if the execution time is 12s, 15s and 22s for machine A, B and C respectively.

Solution: 3 points

$$CPU\ Time = \frac{Clock\ Cycles}{Clock\ Rate}$$

$$\text{Clock Cycles} = \sum_{i=1}^{n} (\text{CPI}_i \times \text{Instruction Count}_i)$$

$Clock\ Rate\ for\ A = \frac{(12\times1)+(8\times2)+(5\times3)+(9\times4)+(6\times5)+(22\times3)}{12} = 14.58\text{HZ}$

$Clock\ Rate\ for\ B = \frac{(11\times1)+(7\times2)+(4\times3)+(7\times4)+(5\times5)+(12\times3)}{15} = 8.4\text{HZ}$

$Clock\ Rate\ for\ C = \frac{(9\times1)+(5.6\times2)+(7\times3)+(12\times4)+(9\times5)+(17\times3)}{22} = 8.42\text{HZ}$

(c) Calculate the MIPS for each machine.

Solution: 3 points

$MIPS\ rate = \frac{I_c}{Execution\ Time \times 10^6}$ or $\frac{f}{CPI \times 10^6}$

$MIPS_A = \frac{14.58}{9.72 \times 10^6} = 1.5 \times 10^{-6}$

$MIPS_B = \frac{8.4}{7 \times 10^6} = 1.2 \times 10^{-6}$

$MIPS_C = \frac{8.42}{10.29 \times 10^6} = 8.16 \times 10^{-7}$

Problem 3. Two processors *simplex* and *vertex* run the same program, with the same input under identical conditions. The program running on vertex takes 30% less time but incurs 25% more CPI compared to the same program running on simplex. If the clock rate of simplex is 5MHz, then:

(a) Calculate the frequency of vertex.

Solution: 6 points

The number of instructions on both machines is identical, therefore:

$\frac{t_s f_s}{CPIs} = \frac{t_v f_v}{CPIv}$

$\frac{t_s \times 5MHZ}{CPI_s} = \frac{0.7 t_s \times f_v}{1.25 CPIs}$

$f_v = \frac{5 \times 1.25}{0.7} = 8.93$ MHZ

Problem 4. Given the following assembly language code for a program starting at memory address 300.

300	LOAD	M(200)
	SUB	M(201)
301	JUMP+	M(303,0:19)
	LOAD	M(201)
302	STORE	M(202)
	JUMP	M(0,0:15)
303	LOAD	M(200)
	STORE	M(202)
304	JUMP	M(0,0:19)

(a) Show the memory contents for an IAS computer.

Solution: 5 points – 1 point for each row

Address	Contents
12C	010C8060C9
12D	0F12F010C9
12E	210CA0D000
12F	010C8210CA
130	0D000

(b) Explain what this program does.

Solution: 2 points

The program finds the maximum of two integers stored in location 200 and 201, and stores the result in location 202.

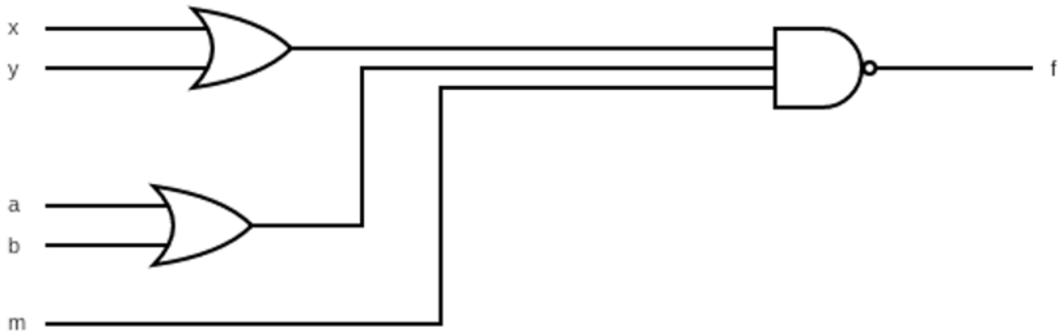
Problem 5. Implement the function f (w1, w2, w3) = $\sum$ m(0, 1, 3, 4, 6, 7) by using NOT, AND and OR gates.

Solution: 5 points — 3 points for deriving the function, 2 points for the circuit diagram

$f(w1, w2, w3) = \overline{w1w2} + w2w3 + w1\overline{w3}$
Many others are possible

(circuit diagram is obvious)

Problem 6. In standard cell technology, circuits are built by interconnecting building-block cells that implement simple functions, like basic logic gates. One type of standard cell is the and-or-invert (AOI) cells. Consider the or-and-invert (OAI) cells which can be efficiently built as CMOS complex gates as shown in the Figure below.



(a) State the function this cell implements.

Solution: 1 point

$f = \overline{(x+y)\cdot(a+b).m}$

(b) Derive the CMOS complex gate that implements this cell.

Solution: 3 points

Applying Demorgan's theorem in two steps gives:

$f = \overline{(x+y)}\cdot\overline{(a+b)}.\overline{m}$

$f = \overline{x}\overline{y} + \overline{a}\overline{b} + \overline{m}$

Problem 7. The Figure below depicts the conversion between three-bit binary and Gray codes. The Gray code is one in which consecutive valuations differ in one variable only.

b_2	b_1	b_0	g_2	g_1	g_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

(a) Find the canonical sum-of-products expressions for g_0, g_1 and g_2.

Solution: 3 points

$g_0 = \overline{b_2 b_1} b_0 + \overline{b_2} b_1 \overline{b_0} + b_2 \overline{b_1} b_0 + b_2 b_1 \overline{b_0}$

$g_1 = \overline{b_2} b_1 \overline{b_0} + \overline{b_2} b_1 b_0 + b_2 \overline{b_1 b_0} + b_2 \overline{b_1} b_0$

$g_2 = b_2 \overline{b_1 b_0} + b_2 \overline{b_1} b_0 + b_2 b_1 \overline{b_0} + b_2 b_1 b_0$

(b) Find the canonical product-of-sums expressions for g_0, g_1 and g_2.

Solution: 3 points

$g_0 = (\overline{b_2} + \overline{b_1} + \overline{b_0}) \cdot (\overline{b_2} + b_1 + b_0) + (b_2 + \overline{b_1} + \overline{b_0}) \cdot (b_2 + b_1 + b_0)$

$g_1 = (\overline{b_2} + \overline{b_1} + \overline{b_0}) \cdot (\overline{b_2} + \overline{b_1} + b_0) + (b_2 + b_1 + \overline{b_0}) \cdot (b_2 + b_1 + b_0)$

$g_2 = (\overline{b_2} + \overline{b_1} + \overline{b_0}) \cdot (\overline{b_2} + \overline{b_1} + b_0) + (\overline{b_2} + b_1 + \overline{b_0}) \cdot (\overline{b_2} + b_1 + b_0)$