CPSC 3300 - Exam 1 Sample Questions

1. Matching -- technology/performance terms. Write the correct term from the

list into each blank. (1 pt. each)

datapath Dennard scaling throughput supercomputer

CPU time Moore’s law bit workload

clock frequency CPI byte Whetstone

a. \_**Moore’s law**\_\_\_\_\_ the observation that the number of transistors in a dense

integrated circuit doubles approximately every two years

b. \_**supercomputer**\_\_\_ a computer with the highest performance and cost

c. \_**bit**\_\_\_\_\_\_\_\_\_\_\_\_\_ a single binary digit (1 or 0)

d. \_**clock frequency**\_ 1/clock cycle time. Also number of clocks per second.

e. \_**CPU time**\_\_\_\_\_\_\_\_ time spent processing a given job

f. \_**throughput**\_\_\_\_\_\_ measure of work done per unit time

2. Give the power of 10 associated with these prefixes. (1 pt. each)

milli \_**10^-3**\_\_ mega \_**10^6**\_\_\_

exa \_**10^18**\_\_ micro \_**10^-6**\_\_

3. For the following workload and cycle values, find the average CPI. (4 pts.)

type | freq cycles

-------+-------------- CPI = \_**1.5**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

alu | 0.70 1

ld/st | 0.20 2

branch | 0.10 4

.7 \* 1 + .2 \* 2 + .1 \* 4 = 1.5

4. Given the workload and cycle values from question 3 with an instruction count of

40 million and a clock frequency of 2 GHz, find the execution time. (8 pts.)

**0.03 seconds IC \* CPI \* Cyc = 40 mil \* 1.5 / 2 bil = 0.03 s**

5. Consider the execution times of two different programs in a benchmark suite on

three different computers:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Computer A | Computer B | Computer C |
| Program 1 | 10 | 12 | 16 |
| Program 2 | 10 | 3 | 4 |

Calculate the normalized execution times of the benchmark suite on the three

computers (using geometric mean) with respect to Computer A: (8 pts.)

A: **1** B: **0.6** C: **0.8**

GM(A) = sqrt(10\*10) = sqrt(100) = 10 norm -> 1

GM(B) = sqrt(12\*3) = sqrt(36) = 6 norm -> 0.6

GM(C) = sqrt(16\*4) = sqrt(64) = 8 norm -> 0.8

6. Give a circuit implementation for an XOR circuit using AND, OR, and NOT

gates. (8 pts.)

a b | XOR

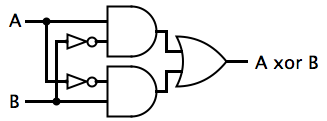
-------+-------------

0 0 | 0

**0 1 | 1 🡪 notA \* B**

**1 0 | 1 -> A \* notB = notA \* B + A \* notB**

1 1 | 0 AND OR AND



7. Consider A + B\*C = (A+B)\*(A+C). Show by truth table that this is true. (10 pts.)

A B C | (B\*C) | A + B\*C | (A+B)\*(A+C) | A+B | A+C

-----------+---------+-------------+---------------+---------+---------

0 0 0 | 0 | 0 | 0 | 0 | 0

0 0 1 | 0 | 0 | 0 | 0 | 1

0 1 0 | 0 | 0 | 0 | 1 | 0

0 1 1 | 1 | 1 | 1 | 1 | 1

1 0 0 | 0 | 1 | 1 | 1 | 1

1 0 1 | 0 | 1 | 1 | 1 | 1

1 1 0 | 0 | 1 | 1 | 1 | 1

1 1 1 | 1 | 1 | 1 | 1 | 1

8. For the Boolean function O1 and O2, as given in the following truth table:

(6 pts. each)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | | | **Output** | |
| **x** | **y** | **z** | **O1** | **O2** |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |

(a) List the minterms for a three-variable function with variables x, y, and z.

**~x\*~y\*~z ~x\*~y\*z ~x\*y\*~z ~x\*y\*z x\*~y\*~z x\*~y\*z x\*y\*~z x\*y\*z**

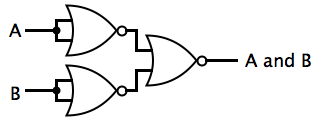
(b) Express O1 and O2 in sum-of-product algebraic form.

**O1 = ~x\*~y\*~z + ~x\*y\*z + x\*~y\*z + x\*y\*~z**

**O2 = ~x\*y\*~z + ~x\*y\*z + x\*~y\*~z + x\*y\*z**

9. You are provided with NOR gates only. (4 pts. each)

(a) draw the circuit to implement the AND function



(b) draw the circuit to implement the OR function



10. Prove the identity of each of the following Boolean equations, using algebraic

manipulation: (4 pts. each)

(a) ~(A \* B) \* (~A + B) \* (~B + B) = ~A

(~A +~B) \* (~A + B) \* (~B + B) = ~A

(~A +~B) \* (~A + B) \* (1) = ~A

(~A +~B) \* (~A + B) = ~A

~A + (~B\*B) = ~A

~A + (0) = ~A

(b) (A \* B) + (B \* C) \* (B + C) = B \* (A + C)

(A \* B) + (B\*C\*B + B\*C\*C) = B \* (A + C)

(A \* B) + (B\*C + B\*C) = B \* (A + C)

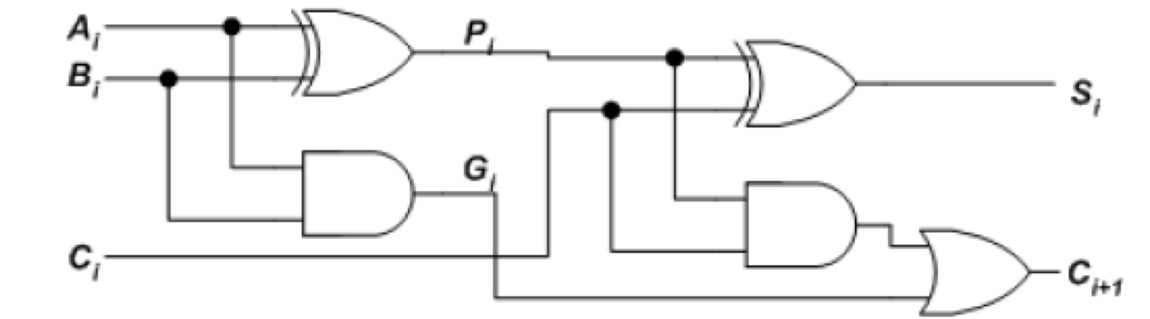
(A \* B) + (B \* C) = B \* (A + C)

(B \* A) + (B \* C) = B \* (A + C)

11. In a carry-lookahead adder two signals are created for each bit position -

P: “(carry) propagate” and G: “(carry) generate” which are used to determine

the carry bit for the next (higher) bit position. (6 pts. each)



(a) Given the formula for Ci+1 (carry bit for next position) is:

Ci+1 = Gi + (Pi \* Ci), write out the formulas for C1, C2 and C3 in terms

of C0, P0, G0, P1, G1, P2 and G2.

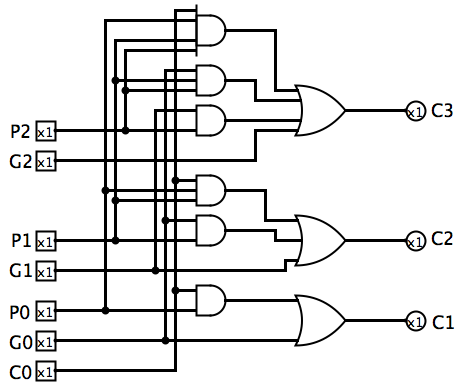
C1 = G0 + P0\*C0

C2 = G1 + P1\*G0 + P1\*P0\*C0

C3 = G2 + P2\*G1 + P2\*P1\*G0 + P2\*P1\*P0\*C0

(b) Draw the circuit diagram of a 3-bit carry-lookahead unit with inputs

C0, P0, G0, P1, G1, P2 and G2 and outputs C1, C2 and C3.



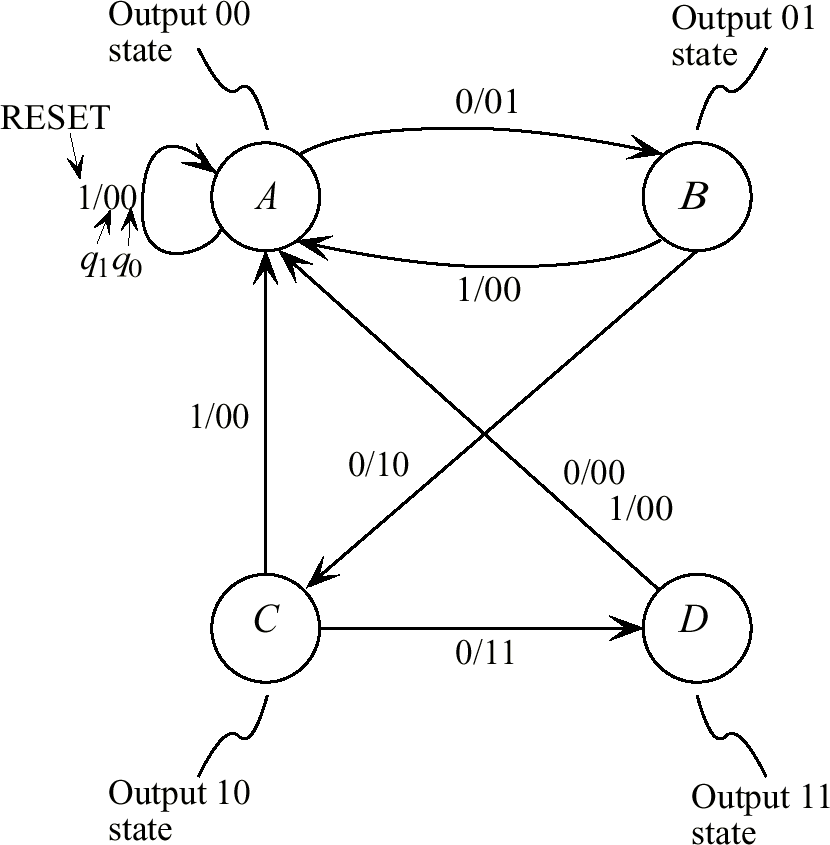
12. Consider a mod-4 counter with input R. When R=0 the counter will increment in a

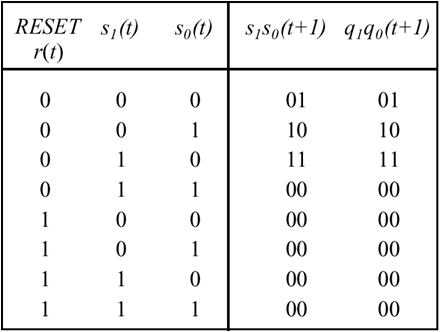
loop from 00 to 11, e.g. 00->01->10->11->00... When R=1 the counter is reset

to 00. (4 pts. each)

(a) Draw the state transition diagram. Label the up transitions with '0' and

the reset transitions with '1'.



 (b) Give the state table using S1(t) and S0(t) for current state.

(c) Give the logic expressions for Q1(t+1) and Q0(t+1).

