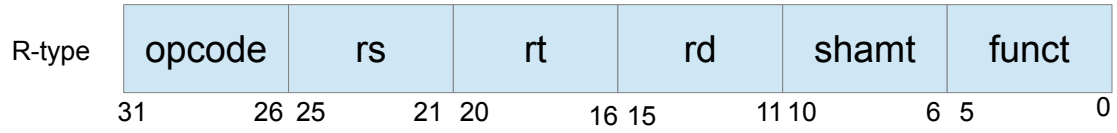
Student_ID: _____

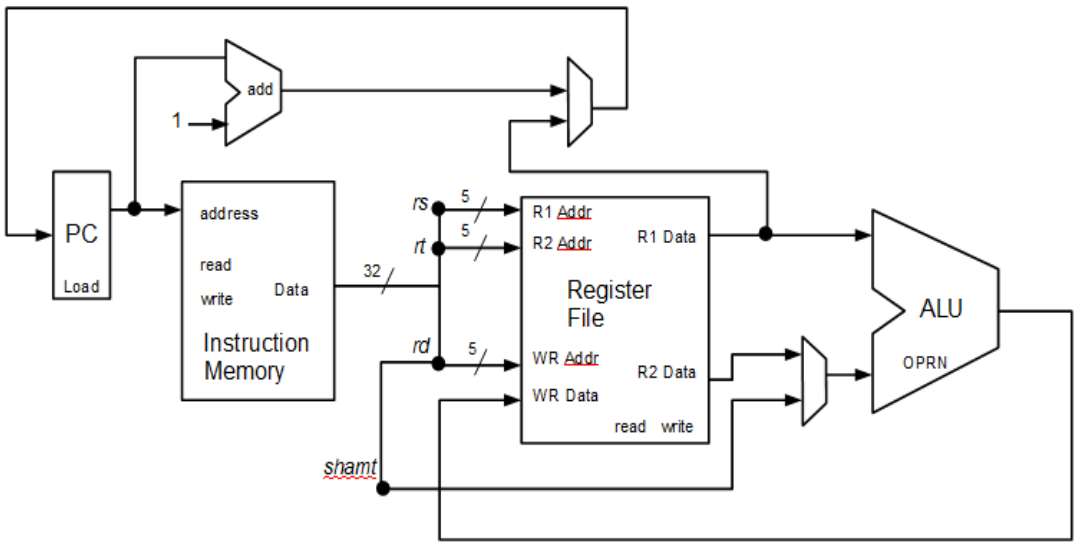
Name	_____
Student ID	_____
Exam	Final Examination CS147, Sec 01
Date	05/15/15 (Time 2 hr. 15 min.)
Instructions	1. There are 6 questions (#1-6) with 5 points each. 2. You need to answer total 20 points or more. 3. Use also the back of the pages for your answer if needed.

DO NOT TURN THE PAGE UNTIL TOLD

1. Draw data path schematic for R-type of instructions supporting CS147DV R-type instructions. Remember that most of the R-type instructions are similar (R[rd] = R[rs] <op> R[rt]) except for SHIFT (where one operand is picked from the instruction encoding field) and JR (Jump register where next PC value is determined by the register file content of the given register RS). The following is the general instruction format for the R-type instructions. [**5pts**]



Ans:



2. A processor implements dynamic frequency scaling with two modes of operation. One is 'turbo' mode at which the processor runs at 1.5GHz and other is 'normal' mode at which the processor at 1.0GHz. This processor deploys a branch prediction algorithm which has failure probability of 0.2. A branch prediction failure incurs 5 cycle of penalty. This processor also deploys a cache mechanism which has hit rate (for both instruction and data access) of 0.7 at turbo mode and 0.9 at normal mode. Any miss will incur 100 cycle of penalty. This system is running a program with 15% conditional branch instruction and 35% data memory access instruction mix. Which mode will run faster and by how much (determine the ratio between the total execution time in two different modes) if average CPI is 3.5? [**5pts**]

Ans:

Total number of cycle in turbo mode is

$$C_{total} = C_{ideal} + C_{branch\ failure} + C_{cache\ miss\ penalty\ for\ instruction} + C_{cache\ miss\ penalty\ for\ data}$$

Total cycle is sum of cycle for ideal execution, total penalty cycle for branch prediction failure, total penalty for cache miss for instruction fetch, total penalty for cache miss for data fetch. Therefore total cycle needed for turbo and normal mode for a program with n instruction is as following.

$$\begin{aligned} C_{turbo} &= 3.5n + (0.15 * n * 0.2 * 5) + (0.3 * n * 100) + (0.3 * 0.35 * n * 100) \\ &= 3.5n + 0.15n + 30n + 10.5n \\ &= 44.15n \end{aligned}$$

$$\begin{aligned} C_{normal} &= 3.5n + (0.15 * n * 0.2 * 5) + (0.1 * n * 100) + (0.1 * 0.35 * n * 100) \\ &= 3.5n + 0.15n + 10n + 3.5n \\ &= 17.15n \end{aligned}$$

Therefor total execution time in two mode will be as following where T is the time period of each cycle and F is the frequency.

$$E_{turbo} = T_{turbo} * C_{turbo} = \frac{C_{turbo}}{F_{turbo}}; \qquad E_{normal} = T_{normal} * C_{normal} = \frac{C_{normal}}{F_{normal}}$$

Hence the performance comparison would be as following.

$$\begin{aligned} \frac{P_{normal}}{P_{turbo}} &= \frac{E_{turbo}}{E_{normal}} = \frac{C_{turbo} \times F_{normal}}{C_{normal} \times F_{turbo}} \\ &= \frac{44.15n \times 1.0GHz}{17.15n \times 1.5GHz} \\ &= \frac{44.15}{25.725} \\ &= 1.72 \end{aligned}$$

Therefore with this program characteristics normal mode is 1.72x faster that turbo mode.

3. Regarding parallel processing systems, answer the following questions.

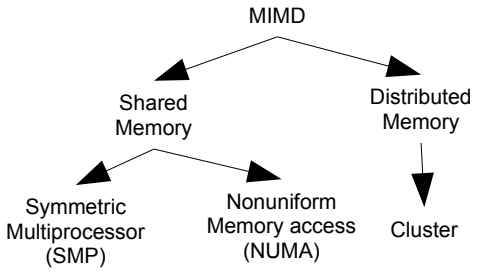
   (a) Show broad classification of different types of MIMD system (Multiple Instruction Multiple Data). [**1pts**]

   (b) Show generic block diagram of a SMP (shared memory symmetric multiprocessor) system. [**2pts**]

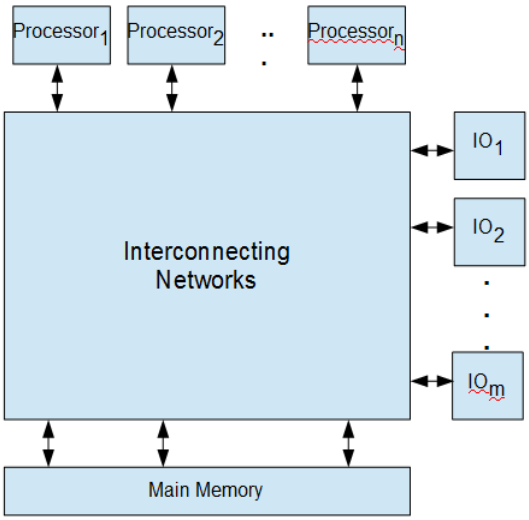
   (c) What is the difference between NUMA system from SMP system? [**2pts**]

Ans:

(a)



(b)



(c) SMP is purely shared memory system, where NUMA is hybrid of shared memory system and distributive memory system. NUMA system can access it's local memory as well as remote memory to other NUMA system through network.

4. A 32-bit word addressable memory system (1 word == 4 bytes per address) deploys both virtual memory and cache. The virtual memory has page size of 4K words (1K=1024). The cache has 4K lines with block size of 256 words and 2 control bits (valid and dirty). The TLB (Translate Look aside Buffer) has two mapping entries as '0x28F67 → 0x287A' and '56764 → 0x1843'.

   (a) What is the total size of the cache memory? [**1pts**]

   (b) Determine the TAG, Cache Index, and block offset in HEX number format for the following virtual addresses.

      i. 0x28F679A2 [**2pts**]

      ii. 0x56764922 [**2pts**]

Ans:

For the virtual memory, Number of bits required to index 4K words is 12. Therefore page number is represented in the system with (32-12)=20 bits.

For the cache, number of bits for block indexing is 8 (for 256 words). The number of cache index bit is 12. Therefor the number of tag bits are (32-12-8)=12.

(a) Total size of cache in terms of bits is ((256*32)+12+2)*4*1024 = 33611776 = 4201472Bytes = 4103KB

(b) (i) Page number of virtual address 0x28F679A2 is 0x28F67. It is mapped to 0x287A. Therefore the physical address is translated to 0x287A9A2. Hence the tag, cache index and block offset is as following.

- Tag : 0x28
- Cache Index : 0x7A9
- Block Offset : 0xA2

(ii) Page number of virtual address 0x56764922 is 0x56764. It is mapped to 0x1843. Therefore the physical address is translated to 0x1843922. Hence the tag, cache index and block offset is as following.

- Tag : 0x18
- Cache Index : 0x439
- Block Offset : 0x22

[***You really do not need to go through the formula and still get the answer since all the page, block, cache lines are in form of 2^n in this case – MOD and DIV operation is just chopping off the hex numbers in the correct position – For example Page number and page index values for 0x56764922 (0x56764|922) is 0x56764 and 0x922 since page size is 4K which is addressed by 12 bits in LSB position, which is the last 3 hex digit, rest 5 hex digit is the page number. Similarly for tag, cache index and block offset we can determine the number directly from the hex. In this case block offset is represented with 8 bits (last 2 hex digtis), page index is represented by middle 12 bits (or middle 3 hex digits) and tag is first 12 bits (or 3 hex digits at the beginning, with the implicit 0). For example 0x1843922 == 0x01843922 (0x018|439|22) has tag of 0x018, cache index of 0x439 and block offset as 0x22]***

5. Related to Boolean algebra answer the following question.

   (a) Describe DeMorgan's identity formula. [**1pts**]

   (b) Simplify the Boolean expression X'YZ + X'Z + X'YZ' [**2pts**]

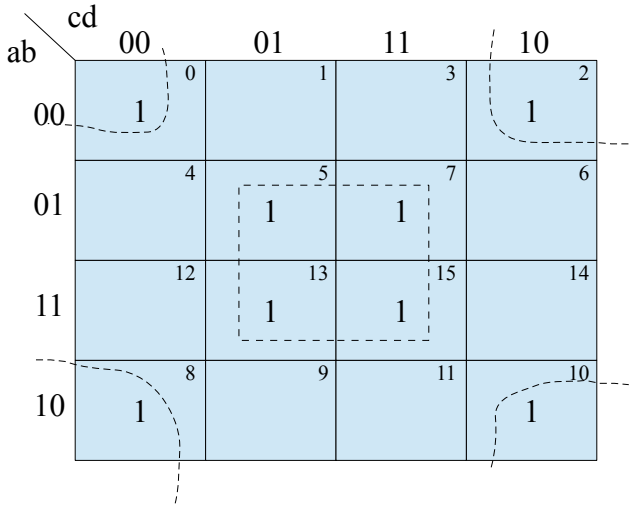
   (c) Simplify the Boolean expression using K-map technique: F(a,b,c,d) = Σm(0,2,5,7,8,10,13,15) [**2pts**]

Ans:

(a) (X+Y)' = X'Y'; (XY)' = X'+Y'

(b) X'YZ + X'Z + X'YZ' = X'Y(Z+Z') + X'Z = X'Y + X'Z = X'(Y+Z)

(c)



f(a,b,c,d) = bd + b'd' = (b ⊕ d)'

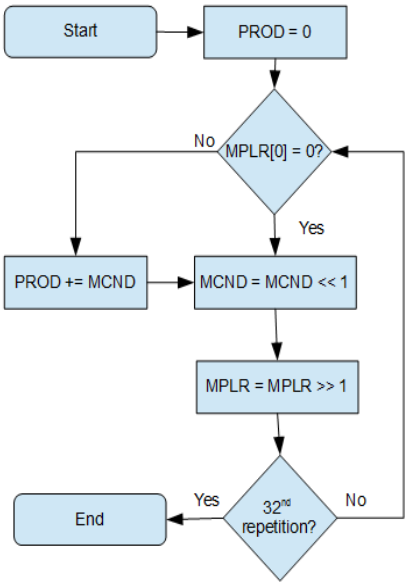
6. Regarding multiplication answer the following questions.

   (a) Draw flow chart for the sequential algorithm of multiplication. [**2pts**]

   (b) Draw schematic for a sequential multiplier. [**3pts**]

Ans:

(a)



(b)