| | | |
|---|---|---|
| **Zagazig University** | **CSE 321a** | **Midterm Examination** |
| **Faculty of Engineering** | **Computer Organization (I)** | **December 13th, 2016** |
| **Computer & Systems Engineering** | **(Double-Sided)** | **11:00am – 12:15pm** |
| **3rd Year – Fall 2016** | **(Duration: 75 minutes)** | **4 pages, 25 questions, 25 points** |

| | | |
|---|---|---|
| **الاسم:** | **القسم:** | **رقم الجلوس:** |

*Circle the letter of the choice that best answers each of the following questions. No more than one letter should be circled.*

1. Which of the following terms refers to an architectural attribute of the computer?
   (a) Memory block
   **(b) Memory addressable unit**
   (c) Memory transfer unit
   (d) Only (a) and (b)
   (e) Only (b) and (c)

2. Which of the following terms describes the operation of individual computer components?
   (a) Computer architecture
   **(b) Computer function**
   (c) Computer structure
   (d) Computer organization
   (e) None of the above

3. Which of the following parameters may change if a high-level program is recompiled differently?
   (a) Average CPI
   (b) Instruction count
   (c) SPEC benchmark score
   **(d) All the above**
   (e) None of the above

4. Suppose the execution of two different program fractions $x_1$ and $x_2$ can be sped up using two different optimizations. What is the maximum speedup achieved by these two optimizations?
   **(a) $\mathbf{1/(1-x_1-x_2)}$**
   (b) $1/(1-max(x_1, x_2))$
   (c) $1/(1-min(x_1, x_2))$
   (d) $1/(1-(x_1+x_2)/2)$
   (e) None of the above

5. A physically-dedicated bus can transfer *w* address bits and *x* data bits each second. The bus supports read-after-write operations in which addresses are *y*-bit long and data are 32-bit long. How much time is taken to complete one read-after-write operation from a memory whose access time is *z* seconds?
   (a) $z+y/w+64/x$ seconds
   (b) $z+y/w+32/x$ seconds
   (c) $\mathbf{z+max(y/w, 32/x)+32/x}$ **seconds**
   (d) $z+max(y/w, 32/x)$ seconds
   (e) None of the above

6. In traditional hierarchical bus architecture, which bus **is not** part of the link between CPU and MM?
   (a) Local bus
   (b) System bus
   **(c) Expansion bus**
   (d) All the above
   (e) None of the above

7. What do associative and random memories have in common?
   (a) Both have a shared read/write mechanism
   **(b) In both cases, access time is independent of location of previous access**

(c) In both cases, data is located by a comparison against a tag
(d) All of the above
(e) None of the above

8. Which of the following memory characteristics captures the main difference between RAM and ROM?
**(a) Volatility**
(b) Location
(c) Access method
(d) Physical type
(e) None of the above

9. Which of the following performance metrics **does not** depend on the recovery time?
**(a) Access time**
(b) Cycle time
(c) Transfer time
(d) All the above
(e) None of the above

10. Which of the following programming habits **may not improve** the hit ratio of the instruction cache?
(a) Using loops
(b) Using subroutines
**(c) Using arrays**
(d) All the above
(e) None of the above

11. Which of the following optimizations **may worsen** the average access time of a memory system?
(a) Increasing the cache line size
(b) Increasing the cache size
(c) Increasing the cache associativity
**(d) All the above**
(e) None of the above

12. Which of the following cache configurations achieves the best load-balancing with multiple CPU's?
(a) Split cache
(b) Dedicated cache
**(c) Shared cache**
(d) Unified cache
(e) None of the above

**Questions 13 – 15 are based on the following information:**

A given processor has eight interrupt lines (numbered 0-7), and a policy that low-numbered interrupts have priority over higher-numbered ones. The processor takes 10 time units to handle an interrupt. Suppose the processor starts with no interrupts pending, and the following sequence of interrupts occurs:

| **Interrupt Number** | 4 | 7 | 1 | 3 | 0 |
|---|---|---|---|---|---|
| **Occurrence Time** | 3 | 5 | 9 | 15 | 22 |

Consider the way interrupts are handled by the processor in the following two cases:
- **Case #1**: interrupts **get disabled** during the execution of the interrupt handlers.
- **Case #2**: interrupts **stay enabled** during the execution of the interrupt handlers.

13. In which order does the processor finish up executing the interrupt handlers in **Case #1**?
(a) 0 → 1 → 3 → 4 → 7
(b) 1 → 0 → 3 → 4 → 7
**(c) 4 → 1 → 0 → 3 → 7**
(d) 4 → 7 → 1 → 3 → 0
(e) None of the above

14. In which order does the processor finish up executing the interrupt handlers in **Case #2**?
(a) 0 → 1 → 3 → 4 → 7

**(b) <u>1 → 0 → 3 → 4 → 7</u>**
(c) 4 → 1 → 0 → 3 → 7
(d) 4 → 7 → 1 → 3 → 0
(e) None of the above

15. Which points in time mark the start and the end for handling interrupt 3 in **Case #2**?
**(a) <u>19 and 39</u>**
(b) 19 and 29
(c) 15 and 35
(d) 15 and 25
(e) None of the above

**Questions 16 – 20 are based on the following information:**

Consider a small computer in which memory locations, general-purpose registers, and machine instructions are all 18-bit long. The processor has only two general-purpose registers (numbered 0 and 1). The five most-significant bits of each machine instruction ($X_{17-13}$) represent an opcode. The remaining bits ($X_{12-0}$) specify two operands:

- First operand (specified by $X_{12-11}$): if $X_{12}$ is 0, the operand value is in register $X_{11}$, otherwise, the operand value is in the memory location whose address is in register $X_{11}$.
- Second operand (specified by $X_{10-0}$): if $X_{10}$ is 0, the operand value is $X_{9-0}$, otherwise, the operand value is the memory location whose address is $X_{9-0}$.

Operand values are interpreted as signed integers according to the sign-and-magnitude representation. The following table contains some of the supported opcodes:

| Mnemonic | Binary | Meaning |
|---|---|---|
| MOVE | 00010 | Copy the value of the second operand to the first operand. |
| SWAP | 00100 | Exchange the values of the two operands. |
| SUB | 01111 | Subtract the (signed) value of the first operand minus the (signed) value of the second operand and save the result to the first operand. |
| DECBRNEG | 10010 | Decrement the (signed) value of the first operand by 1, and if the new value is negative, branch to the instruction whose address is obtained by adding the (signed) value of the second operand to the PC, else continue normally. |

Suppose the values of the registers and memory locations in the **initial state** are:

| PC | MAR | Register 0 | Register 1 | Location 0019C | Location 0019D |
|---|---|---|---|---|---|
| 02ABE | 0BEEF | 3CA35 | 15E1D | 2D24A | 21124 |

16. What is the maximum range of signed values that can be taken by the second operand?
(a) $-2^{18}-1 \rightarrow 2^{18}-1$
**(b) <u>$-2^{17}-1 \rightarrow 2^{17}-1$</u>**
(c) $-2^{10}-1 \rightarrow 2^{10}-1$
(d) $-2^{9}-1 \rightarrow 2^{9}-1$
(e) None of the above

17. How many memory accesses are needed to fetch and execute instruction 096BE?
(a) 1 (b) 2 (c) 3 (d) 4 **(e) <u>None of the above</u>**

18. What will be the value of MAR after executing instruction 05D9C from initial state?
(a) 02ABF
**(b) <u>15E1D</u>**
(c) 0019C
(d) 2D24A
(e) None of the above

19. What will be the value of register 0 after executing instruction 1E59C from initial state?
(a) 3CA35

(b) 29C7F
(c) 1C899
**(d) <u>2F7EB</u>**
(e) None of the above

20. Which of the following instructions will load PC with 0199A after being executed from initial state?
(a) 2419D
(b) 2499D
**(c) <u>2459D</u>**
(d) 24D9D
(e) None of the above

**Questions 21 – 25 are based on the following information:**

A computer has a 1 MB main memory (that is byte-addressable) and a *k*-way set-associative cache. The following table symbolically describes the first nine memory references (*i.e.*, read/write operations) made during the execution of a program (assuming the cache was initially empty):

| Address (decimal) | 613162 | 744236 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **R**ead/**W**rite (R/W) | R | R | R | W | R | R | R | R | W |
| Word | *x* | 2*x* | | | | | | | |
| Set | *y* | *y* | *y*+9 | *y* | *y* | *y* | *y*+9 | *y* | *y* |
| Tag | *z* | *z*+8 | *z*+8 | *z*+5 | *z*+8 | *z*+5 | *z*+8 | *z* | *z*+3 |
| **H**it/**M**iss (H/M) | M | M | M | M | H | H | H | H | M |
| Write to MM? (Y/N) | N | N | N | N | N | N | N | N | Y |

21. What is the format of the memory address?
(a) Tag → 3 bits, Set → 15 bits, Word → 2 bits
(b) Tag → 3 bits, Set → 14 bits, Word → 3 bits
(c) Tag → 7 bits, Set → 9 bits, Word → 4 bits
**(d) <u>Tag → 6 bits, Set → 11 bits, Word → 3 bits</u>**
(e) None of the above

22. What is the value of *k*?
(a) 1 (b) 2 **(c) <u>3</u>** (d) 4 (e) None of the above

23. What is the size of the cache?
(a) 2 KB (b) 16 KB **(c) <u>48 KB</u>** (d) 64 KB (e) None of the above

24. Which replacement algorithm **is definitely not** used in this cache?
**(a) <u>LRU</u>**
(b) LFU
(c) Random
(d) All the above
(e) None of the above

25. Which write policy is used in this cache?
(a) No-write-allocate, write-through
(b) Write-allocate, write-through
(c) No-write-allocate, write-back
**(d) <u>Write-allocate, write-back</u>**
(e) None of the above

** End of Exam **