

**MIDTERM EXAMINATION**  
**Spring 2010**  
**CS501- Advance Computer Architecture (Session - 5)**

**Time: 60 min**  
**Marks: 38**

**Question No: 1 ( Marks: 1 ) - Please choose one**

**What is the instruction length of the SRC processor?**

- ▶ 8 bits
- ▶ 16 bits
- ▶ 32 bits
- ▶ 64 bits

**Question No: 2 ( Marks: 1 ) - Please choose one**

**Which one of the following is the memory organization of **FALCON-E** processor?**

- ▶  $2^8 * 8$  bits
- ▶  $2^{16} * 8$  bits
- ▶  $2^{32} * 8$  bits
- ▶  $2^{64} * 8$  bits

**Question No: 3 ( Marks: 1 ) - Please choose one**

**“If  $P = 1$ , then load the contents of register R1 into register R2”.**

**This statement can be written in RTL as:**

- ▶  $R1 \leftarrow R2$
- ▶  $P: R1 \leftarrow R2$
- ▶  $P: R2 \leftarrow R1$
- ▶  $P: R2 \leftarrow R1, P: R1 \leftarrow R2$

**Question No: 4 ( Marks: 1 ) - Please choose one**

The instruction -----will **load** the register R3 with the contents of the memory location M [PC+56] <http://vustudents.ning.com>

- ▶ Add R3, 56
- ▶ lar R3, 56
- ▶ **ldr R3, 56**
- ▶ str R3, 56

**Question No: 5 ( Marks: 1 ) - Please choose one**

-----are faster than cache memory <http://vustudents.ning.com>

- ≡ Accumulator register
- ≡ CPU registers
- ≡ I/O devices
- ≡ ROM

**Question No: 6 ( Marks: 1 ) - Please choose one**

**P: R3 → R5**

**MAR → IR**

These two are instructions written using RTL .If these two operations is to occur simultaneously then which symbol will we use to separate them so that it becomes a correct statement with the condition that two operations occur simultaneously?

- ▶ Arrow →
- ▶ Colon :
- ▶ **Comma ,**
- ▶ Parentheses ()

**Question No: 7 ( Marks: 1 ) - Please choose one**

Prefetching can be considered a primitive form of-----

- ▶ **Pipelining** <http://vustudents.ning.com>

- . ▶ Multi-processing
- . ▶ Self-execution
- . ▶ Exception

**Question No: 8 ( Marks: 1 ) - Please choose one**

The processor must have a way of saving information about its state or context so that it can be restored upon return from the ----- <http://vustudents.ning.com>

- . ▶ Exception
- . ▶ Function
- . ▶ Stack
- . ▶ Thread

**Question No: 9 ( Marks: 1 ) - Please choose one**

Which one of the following circuit design levels is called the gate level?

- ☒ Logic Design Level
- ☐ Circuit Level
- ☐ Mask Level
- ☐ None of the given

**Question No: 10 ( Marks: 1 ) - Please choose one**

\_\_\_\_\_ enable the input to the PC for receiving a value that is currently on the internal processor bus.

- ▶ LPC
- ▶ INC4
- ▶ LC
- ▶ Cout

**Question No: 11 ( Marks: 1 ) - Please choose one**

\_\_\_\_\_ operation is required to change the processor's state to a known, defined value.

- ▶ Change

- ▶ Reset
- ▶ Update
- ▶ None of the given

**Question No: 12 ( Marks: 1 ) - Please choose one**

There are \_\_\_\_\_ types of reset operations in SRC

- ▶ Two
- ▶ Three
- ▶ Four
- ▶ Five

**Question No: 13 ( Marks: 1 ) - Please choose one**

\_\_\_\_\_ controller controls the sequence of the flow of microinstructions.

- ▶ Multiplexer
- ▶ Microprogram
- ▶ ALU
- ▶ None of the given

**Question No: 14 ( Marks: 1 ) - Please choose one**

FALCON-A processor bus has 16 lines or is 16-bits wide while that of SRC is \_\_\_\_\_ wide.

- ▶ 8-bits
- ▶ 24-bits
- ▶ 32-bits
- ▶ 64-bits

**Question No: 15 ( Marks: 1 ) - Please choose one**

**Which of the following statement(s) is/are correct about Reduced Instruction Set Computer (RISC) architectures.**

- (i) The typical RISC machine instruction set is small, and is usually a subject of a CISC instruction set.
- (ii) No arithmetic or logical instruction can refer to the memory directly.
- (iii) A comparatively large number of user registers are available.
- (iv) Instructions can be easily decoded through hard-wired control units.

- ▶ (i) and (iii) only
- ▶ (i), (iii) and (iv)
- ▶ (i), (ii) and (iii) only
- ▶ (i),(ii),(iii) and (iv)

**Question No: 16 ( Marks: 1 ) - Please choose one**

Which one of the following register holds the instruction that is being executed?

- ▶ Accumulator
- ▶ Address Mask
- ▶ Instruction Register
- ▶ Program Counter

---

**Question No: 17 ( Marks: 2 )**

Write the following statement of an Arithmetic Instruction using RTL.

*If op-code is 0, the instruction is 'add'. The values in register rb and rc are added and the result is stored in register rc*

<http://vustudents.ning.com>

---

**Question No: 18 ( Marks: 2 )**

Given below are the various fields of an SRC instruction register.

- a) operation code field : **op<4..0>**
- b) target register field : **ra<4..0>**
- c) operand, address index, or branch target register : **rb<4..0>**
- d) second operand, conditional test, or shift count register: **rc<4..0>**

Rewrite these various fields of an SRC instruction, using the RTL.

---

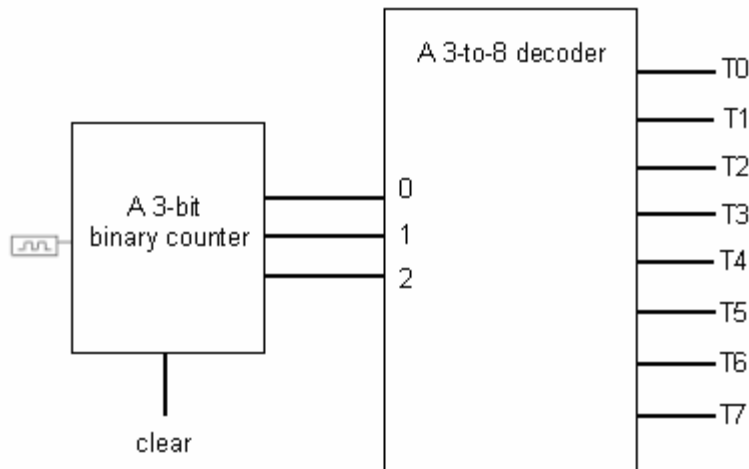
**Question No: 19 ( Marks: 2 )**

How can you define microprogram?

---

**Question No: 20 ( Marks: 3 )**

What is the role of timing step generator in a processor?



**Question No: 21 ( Marks: 3 )**

What is the utility of reset operation and when it is required?

**Question No: 22 ( Marks: 5 )**

Write the **Structural RTL description for un-conditional jump instruction for uni-bus data path implementation.**

**jump [ra+c2]**

**Question No: 23 ( Marks: 5 )**

What function is performed by the reset operation of a processor? What are the two types of reset operations?

2<sup>nd</sup> paper

There were 23 questions 1-14 are mcqs others are questions

Q : DEFINE HARD RESET AND SOFT RESET OPERATIONS IN SRC

Q : Write two pipelining problem and define them briefly.

Q : What information is provided by the addressing modes of some processors?

Q : eLEBORATE PRE-FATCHING CONCEPT?

Q : Write RTL functions and there was a rb +rc instruction.

Q : how we speed-up a computer?

Q : Write execution time of an instruction(there was a description too)

Q : Types of instructions

Q : How you represent register data field?

## CS501-Advance Computer Architecture Midterm Special 2006

Q1

\_\_\_\_\_all memory systems are dumb, in that they respond to only two commands: read or write

Virtually

Logically

Physically

None of These

Q 2

**Consider two programs having three types of instructions given as follows:  
[10]**

**Number of**

**Data transfer instructions**

**Control instructions**

**ALSU instructions**

**Program 1**

7

3

6

**Program 2**

12

5

3

**Instructions Type**

**Control**

**ALSU**

**Data Transfer**

**CPI**

5

3

4

**Compare both the programs for the following parameters:**

**1. Instruction count ( IC )**

**2. Speed of execution ( ET )**

Q 3

To access an operand in memory, the CPU must first generate an address, which it then issues to the \_\_\_\_\_  
MEMORY

REGISTER  
DATA BUS  
ALL OF ABOVE

Q 4

\_\_\_\_\_ or Branch instructions affect the order in which instructions are performed, or control the flow of the program

Control

DATA MOVEMENT

Arithmetic

LOGICAL

Q5

**Reverse assemble the following SRC machine language instructions:**

**[10]**

**68C2003A h**

Q6

An instruction that specifies one operand in memory and one operand in a register would be known as a \_\_\_\_\_ address instruction.

2-1/2

**1-1/2**

0

2

Q7

The data movement instructions \_\_\_\_\_ data within the machine and to or from input/output devices

**Store**

**LOAD**

MOVE

NONE OF ABOVE.

Q8

**Write short answers to the following questions:**

**[3 x 5]**

a. What is the advantage of a linker in the development of assembly language programs?

**Solution:-**

**The linker:**

When developing large programs, different people working at the same time can develop separate modules of functionality. These modules can then be 'linked' to form a single module that can be loaded and executed. The modularity of programs, that the linking step in assembly language makes possible, provides the same convenience as it does in higher-level languages; namely abstraction and separation of concerns. Once the functionality of a module has been verified for correctness, it can be re-used in any number of other modules. The programmer can focus on other parts of the program. This is the so-called "modular" approach, or the "top-down" approach.

b. Define term "Single stepping".

### **Solution:-**

#### **Single stepping:**

Single stepping and breakpoints that allow the examination of the status of the program and registers at desired points during execution.

c. Define term “Type checking”.

### **Solution:-**

#### **Type Checking:-**

High-level languages provide various primitive data types, such as integer, Boolean and a string, that a programmer can use. Type checking provides for the verification of proper usage of these data types. It allows the compiler to determine memory requirements for variables and helping in the detection of bad programming practices.

On the other hand, there is generally no provision for type checking at the machine level, and hence, no provision for type checking in assembly language. The machine only sees strings of bits. Instructions interpret the strings as a type, and it is usually limited to signed or unsigned integers and floating point numbers. A given 32-bit word might be an instruction, an integer, a floating-point number, or 4 ASCII characters. It is the task of the compiler writer to determine how high-level language data types will be implemented using the data types available at the machine level, and how type checking will be implemented.

d. Define term “Instruction set”.

### **Solution:-**

#### **Instruction Set**

A collection of all possible machine language commands that a computer can understand and execute is called its instruction set. Every processor has its own unique instruction set. Therefore, programs written for one processor will generally not run on another processor. This is quite unlike programs written in higher-level languages, which may be portable. Assembly/machine languages are generally unique to the processors on which they are run, because of the differences in computer architecture.

Three ways to list instructions in an instruction set of a computer:

- by function categories
- by an alphabetic ordering of mnemonics
- by an ascending order of op-codes

e. Why computer logic design is different from classical logic design?

### **Solution:-**

#### **Classical logic design versus computer logic design:**

The traditional sequential circuit design techniques for a finite state machine are not very practical when it comes to the design of a computer, in spite of the fact that a computer is a finite state machine. The reason is that employing these techniques is much too complex as the computer can assume hundreds of states.