BUE The British University In Egypt الجامعة البريطانية في مصر	**Examination Paper Proofing & Printing Confirmation Sheet**

Module Title : **Computer Architecture**	Module Code: **13CSCI10I**
Module Leader **Professor Samir Abou El-Seoud**	Semester **One** **Fall 2013**
Proofed by Dr. Abeer Hamdy	Date of examination

I hereby confirm:

That this examination paper assesses the ILOs defined in the module specification	*
That appropriate model answers were provided with this examination paper	*
That this examination paper has been proof-read and is approved for printing	*
That this examination paper follows the approved University template	*

Signed (Proof Reader): *Dr. Abeer Hamdy*

Printing instructions & stationery requirements

Number of copies of examination paper to be printed		
Date of examination		
		Number required per student
Stationery Requirement(s)	8 page answer book	
	12 page answer book	
	Graph paper	
	Other	

Signed (Module Leader) *Prfessor Samir Abou El-Seoud*

BUE The British University In Egypt الجامعة البريطانية في مصر	**13CSCI10I** **Final Examination, 2012-2013**
Module Title **Computer Architecture**	
Module Leader **Professor Samir Abou El-Seoud**	Semester **One Fall 2013**
Equipment allowed (for example calculator)	Only simple calculator and **NOT** scientific one

Instructions to Students

- *You should attempt **all questions** of section **A***
- *Select only **3 questions** from section **B***
- *The exam paper is **4** pages long, and is in **2** sections*
- *The allocation of marks is shown in brackets by the questions.*

This examination is **TWO** hours long.

Section A

Answer All Questions

QA 1

a) State what are the following acronyms stand for? Discuss briefly the function of each.

i. IR

ii. DMA

iii. MAR

iv. MBR

[10 marks]

b) Define briefly each of the following:

i. Execute cycle

ii. Distributed System

iii. Direct Addressing Mode

iv. Microinstruction **[10 marks]**

[Total 20 marks]

QA 2 The 8-bit registers AR, BR, CR, and DR initially have the following values:

AR = 11110011 BR = 11001111 CR = 10111011 DR = 10101110

Determine the 8-bit values in each register after the execution of the following sequence of microoperation:

$AR \leftarrow AR + BR$

$CR \leftarrow CR \oplus DR, BR \leftarrow CR \wedge AR$

$AR \leftarrow AR - CR$

[Total 20 marks]

QA 3

a) What is the difference between vector and parallel processing?

[10 marks]

b) What is the difference between non-pipelined and pipelined processor?

[10 marks]

[Total 20 marks]

QA 4 Assume an instruction has been read from the memory and the fetch cycle is over. At this moment, the last instruction fetched will be held in IR. The next step is to fetch source operands. Assume the CU has found that the IR contains an operand specifier using *indirect addressing.* In this case an indirect cycle must precede the execute cycle.

a) Use a simple diagram to show the data flow during an indirect cycle.

[10 marks]

b) What is the sequence of microoperations that might be used to represent the indirect cycle? **[5 marks]**

c) Explain the effect of each microinstruction. **[5 marks]**

[Total 20 marks]

Section B

Answer Only THREE Questions

QB 1

a) Draw a space-time diagram for a seven-segment pipeline showing the time it takes to process nine tasks. **[5 marks]**

b) A non-pipeline system takes 70 ns to process a given task. The same task can be processed in a 7-segment pipeline with a clock cycle of 10 ns. Determine the speedup ratio of the pipeline to process 250 tasks. What is the maximum speed up that can be achieved. **[5 marks]**

[Total 10 marks]

QB 2

a) Convert the following arithmetic expressions from infix to reverse Polish notation: **[5 marks]**

i. A*B + C*D + E*F

ii. A+B*[C*D+E*(F+G]

b) Consider a 32-bit microprocessor, with a 16-bit external data bus, driven by an 8-MHz input clock. Assume that this microprocessor has a bus cycle whose minimum duration equals four input clock cycles. What is the maximum data transfer rate that this microprocessor can sustain? **[5 marks]**

[Total 10 marks]

QB 3 What is a bus structure? Describe briefly the function of the lines that forming a bus system. **[Total 10 marks]**

QB 4 Write an ARM assembly program that calculates the average of an array of 8 numbers, say {2, 4, 7, 3, 5, 1, 6, 12}, and put the average in R10. The array is stored at the memory address labelled Array_1. **[Total 10 marks]**

.

Model Answer

BUE The British University In Egypt الجامعة البريطانية في مصر	**13CSCI10I** **Final Examination,** **2012-2013**
Module Title **Computer Architecture**	
Module Leader **Professor Samir Abou El-Seoud**	Semester **One** **Fall 2013**
Equipment allowed (for example calculator)	

Instructions to Students

- *You should attempt **all questions** of section **A***
- *Select only **3 questions** from section **B***
- *The exam paper is **4** pages long, and is in **2** sections*
- *The allocation of marks is shown in brackets by the questions.*

This examination is **TWO** hours long.

Model Answer
Section A

Answer ALL Questions

QA 1

a)

i. **Instruction Register (IR):** A register that is used to hold an instruction for interpretation. It is used to store the current instructions which are being executed. (Instruction register: holds the instruction while it is being executed) It holds the last instruction fetched. **[2.5 marks]**

ii. **Direct Memory Address (DMA):** A form of I/O in which a special module, called a DMA module, controls the exchange of data between main memory and an I/O module. The CPU sends a request for the transfer of a block of data to the DMA module and is interrupted only after the entire block has been transferred. **[2.5 marks]**

iii. **Memory address Register (MAR):** *The address on the memory location from which data has to be read is stored here.* **[2.5 marks]**

iv. **Memory Buffer Register (MBR)***:* The data read from the memory location is stored in this registers. **[2.5 marks]**

b)

i. **Execute Cycle**: That portion of the instruction cycle during which the CPU performs the operations specified by the instruction cycle. **[2.5 marks]**

ii. **Distributed System:** A distributed system is a computer system consisting of a collection of autonomous computers linked by a network and equipped with software that enables the computers to coordinate their activities and to share the resources of system hardware, software, and data, so that users perceive a single, integrated computing facility. In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors. **[2.5 marks]**

iii. **Pipeline Processor:** A pipeline processor executes instructions in an assembly line manner so multiple tasks can be performed simultaneously. However, at no given time can two of the same tasks be performed, so each task is allotted the same time as the task that takes the longest amount of time. Think of an automated car wash line, where multiple cars are serviced, but only one vehicle at a time can be shampooed, conditioned or dried. **[2.5 marks]**

iv. **Microinstruction:** Microinstruction is an instruction that controls data flow and sequencing in a processor at a more fundamental level than machine instructions. Individual machine instructions and perhaps other functions may be implemented by microprograms. **Microinstruction** is an instruction that controls data flow and sequencing in a processor at more fundamental level than machine instructions. Individual machine instructions and perhaps other functions may be implemented by microprograms.
A **microinstruction** is a simple command that makes the hardware operates properly. The format is unique to each computer. Microinstruction is an instruction stored in control memory. The format is unique to each computer. Microinstruction is an instruction stored in control memory. **[2.5 marks]**

[Total 20 marks]

QA 2

AR ← AR + BR ≡ **111000010 (**11110011+11001111)

CR ← CR ⊕ DR ≡ **00010101** (10111011 XOR 10101110),

BR ← CR Λ AR ≡ **010000010** (10111011 AND 111000010)

AR ← AR – CR ≡ **110101101** (111000010 – 00010101)

QA 3

a) **Vector processing**

A computer with built-in instructions that perform multiple calculations on vectors (one-dimensional arrays) simultaneously. It is used to solve the same or similar problems as an array processor; however, a vector processor passes a vector to a functional unit, whereas an array processor passes each element of a vector to a different arithmetic unit.

Vector processing is a procedure for speeding the processing of information by a computer, in which pipelined units perform arithmetic operations on uniform, linear arrays of data values, and a single instruction involves the execution of the same operation on every element of the array

A **vector processor**, or **array processor**, is a central processing unit (CPU) that implements an instruction set containing instructions that operate on one-dimensional arrays of data called vectors. This is in contrast to a scalar processor, whose instructions operate on single data items.

Parallel processing

Parallel processing involves a technique by which complex data sets are broken into individual threads and processed simultaneously across one or more cores. Both AMD and Intel processors have incorporated this technique (known as HTT) to greatly increase the speed at which they operate. Until recently, this did not always provide a significant increase in speed because the technology to properly split up data sets and then bring them back together was in its infancy.

Parallel computing is the processing of data many bits at a time as opposed to serial computing which is the processing of data one bit at a time. In other words, parallel computer runs more tasks in **parallel**, over more CPUs, for faster execution.

[10 marks]

b) In a **non-pipelined** processor, each instruction is executed completely before execution of the next instruction begins. In a **non-pipelined** CPU, the scheduler merely chooses from the pool of waiting work each time an execution unit signals it is free.

In a **pipelined** processor, instruction execution is divided into stages and execution of the next instruction starts as soon as the current instruction has completed the first stage. This increases the rate at which instructions can be executed, improving performance.

A pipeline does not speed up an individual computation. A pipeline processor executes instructions in an assembly line manner so multiple tasks can be performed simultaneously. The net effect is that results are output more quickly than in a non-pipelined unit. This increases the **throughput**, the number of results generated per time unit.

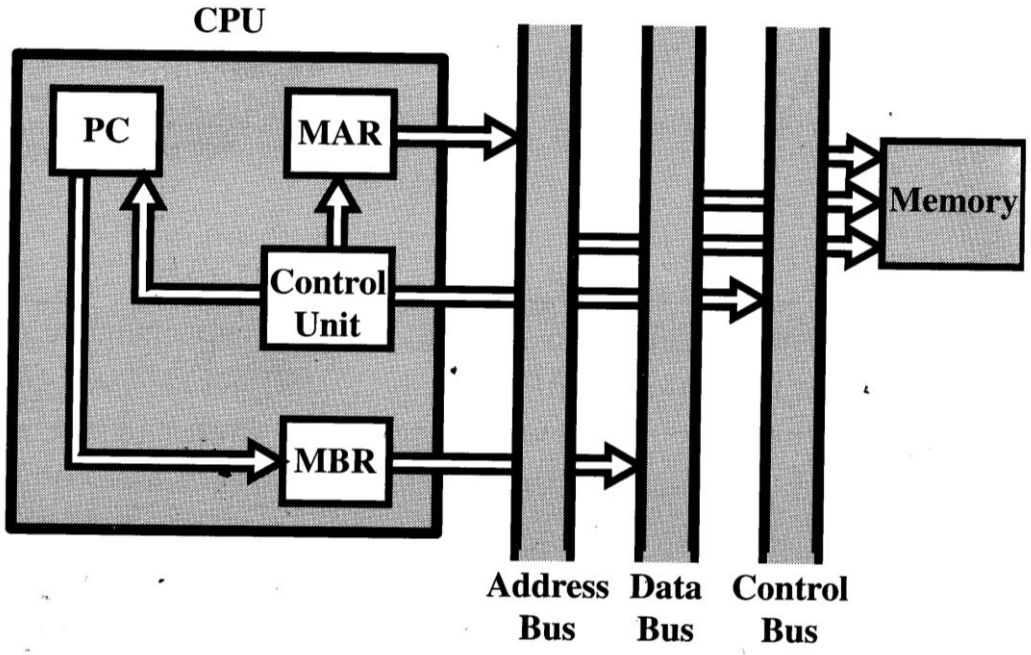
Throughput is also defined to be the rate at which operations get executed (generally expressed as operations/second or operations/cycle).
Pipelining does not reduce the time to complete an instruction, but increases the number of instructions that can be processed at once.

Pipelined CPUs are significantly more efficient than **non-pipelined** CPUs, provided the scheduler can keep the pipeline full. **[10 marks]**

[Total 20 marks]

QA 4

a) Data Flow, Indirect Cycle



[10 marks]

b) Symbolically, we can write this sequence of events as follows:

t_1: MAR ← (IR (Address))
t_2: MBR ←Memory
t_3: IR (Address) ← (MBR (Address))

The above data flow differs somewhat from that indicated in the above diagram. **[5 marks]**

c)

- In the first step, the address field of the instruction stored in the IR is transferred to the MAR.

- In the second step, the address stored in the MAR is then used to fetch the memory address of the operand. The result is placed on the data bus and copied into the MBR

- The final step is to update the address field of the IR from the MBR, so that it now contains a direct rather than an indirect address.

[5 marks]

[Total 20 marks]

Section B

Answer Only THREE Questions

QB 1

c)

Segment	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9						
2		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9					
3			T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9				
4				T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9			
5					T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9		
6						T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	
7							T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9

Therefore, to complete n tasks using a k-segment pipeline requires: k + (n-1).

Hence the time required to complete all operations is 7+9-1 = 15 clock cycles.

[5 marks]

d)

The speedup of a pipeline processing over an equivalent non-pipeline processing may be defined by the ratio:

$$S = nt_n/(k+n-1)t_p$$

Here t_n denotes the time taken by the nonpipeline processor to complete each task and n is the number of tasks to be performed. The total time for the n tasks is nt_n. In the above equation, t_p refers to the clock cycle time used by a k-segment pipeline to execute the n tasks. Note that the time required by the k-segment pipeline to complete the n tasks is k+(n-1) clock cycles.

In our example above, we have t_n = 70 ns, k= 7 segments, t_p= 10 ns, and n=250 tasks.

Therefore

$$S = nt_n/(k+n-1)t_p = (250x70)/[(7+250-1)x10] = 6.836$$

$$S_{max} = t_n / t_p = 60/10 = 6$$

[5 marks]

[Total 10 marks]

QB 2

a)

i. AB * CD *EF *++

ii. FG + E * CD * + * + **[5 marks]**

b) Clock cycle = 1/8 MHz = 125 ns

Bus cycle = 4 x 125 ns = 500 ns

2 bytes transferred every 500 ns; thus transfer rate = 4 Mbytes/sec

[5 marks]

[Total 10 marks]

QB 3 In computer architecture, a bus is a collection of wires through which data is transmitted from one part of a computer to another. You can think of a bus as a highway on which data travels within a computer. When used in reference to personal computers, the term *bus* usually refers to *internal bus or system bus*. This is a bus that connects all the internal computer components to the CPU and main memory. There's also an expansion bus that enables expansion boards to access the CPU and memory. The size of a bus, known as its *width,* is important because it determines how much data can be transmitted at one time. For example, a 16-bit bus can transmit 16 bits of data, whereas a 32-bit bus can transmit 32 bits of data. Every bus has a clock speed measured in MHz. A fast bus allows data to be transferred faster, which makes applications run faster. On PCs, the old ISA bus is being replaced by faster buses such as PCI. Nearly all PCs made today include a local bus for data that requires especially fast transfer speeds, such as video data. The local bus is a high-speed pathway that connects directly to the processor. Several different types of buses are used on Apple Macintosh computers. Older Macs use a bus called NuBus, but newer ones use PCI.

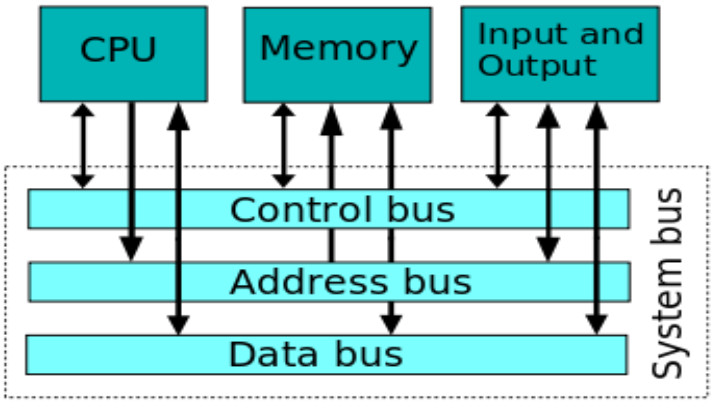
A **system bus** is a single computer bus that connects the major components of a computer system. All system buses consist of three parts – a data bus, an address bus, and a control bus. The data bus transfers actual data whereas the address bus transfers information about where the data should go. The control bus carries commands from the CPU and returns status signals from the devices. The technique was developed to reduce costs and improve modularity. It combines the functions of:

- **A data bus** to carry data between the CPU and its random-access memory. It is an internal pathway across which data are transferred to and from the processor or to and from memory. The width and clock rate (the fundamental rate in cycles per second at which a computer performs its most basic operations such as adding two numbers or transferring a value from one register to another) of the data bus determine its data rate (the number of bytes per second it can carry), which is one of the main factors determining the processing power of a computer. Most current processor designs use a 32-bit bus, meaning that 32 bits of data can be transferred at once. Some processors have an internal data bus which is wider than their external bus in order to make external connections cheaper while retaining some of the benefits in processing power of a wider bus.

- **An address bus** to determine where it should be sent. An **address bus** is a series of lines connecting two or more devices that is used to specify a physical address. When a processor needs to read or write to a memory location, it specifies that memory location on the address bus (the value to be read or written is sent on the data bus). The width of the address bus determines the amount of memory a system can address. For example, a system with a 32-bit address bus can address 232 (4,294,967,296) memory locations. If each memory address holds one byte, the addressable memory space is 4 GB.

- **A control bus** to determine its operation. A **control bus** is used by CPUs for communicating with other devices within the computer. While the address bus carries the information on which device the CPU is communicating with and the data bus carries the actual data being processed, the control bus carries commands from the CPU and returns status signals from the devices, for example if the data is being read or written to the device the appropriate line (read or write) will be active (logic zero).

Modern computers use a variety of separate buses adapted to more specific needs. Most current processor designs use a 32-bit bus, meaning that 32 bits of data can be transferred at once. Some processors have an internal data bus which is wider than their external bus in order to make external connections cheaper while retaining some of the benefits in processing power of a wider bus.



[Total 10 marks]

QB 4

```
                AREA Constants, DATA, READWRITE
                Array_1         DCB 2, 4, 7, 3, 5, 1, 6, 12

                AREA Problem1, CODE, READWRITE
                ENTRY

                MOV R12,#0          ; holds the sum
                MOV R1, #0          ; counter
                LDR R0, =array_1

again           LDRB R4, [R0]       ; read first element.
                ADD R12, R12, R4
                ADD R0, R0, #1      ; point to the next element
                ADD R1, R1, #1      ; increment the counter
                CMP R1, #8
                BLT again

                MOV R10, R12, LSR #3

                END
```

[Total 10 marks]

Module Code: 13CSCI10I	**Title:** Computer Architecture
Modular weight: 10	**Examination weighting:** 60%
Prerequisite modules: CSCI08C	
Reassessment: No restriction.	
Module Leader: Prof. Samir Elsaoud	
Semester taught: 1	
Key words: CPU Architecture, Machine Instructions, Hardware Design, Memory Organization, Structured Design Methodology, Assembler, Assembly Language Programming	
Date of latest revision: Aug. 2013	

Aims

The module aims to impart fundamental knowledge of modern computer architectures in terms of instruction set architecture, organisation and hardware. It develops an understanding of the architectural features and the principles of operation of modern microprocessors and peripheral devices. The module further seeks to provide a sound understanding in the following:

The main families of microprocessors and their differences;
How computers execute their programmes at machine instruction level; and
Principles of the practical design of processor architectures and how design features influence machine coding and performance features.

Intended Learning Outcomes

Upon successful completion of this module students should be able to demonstrate ability in:

Knowledge and understanding

1. To develop and demonstrate knowledge and understanding, abilities and skills in functionality of computer architecture and organization[5]

Subject-specific skills

2. Conceptualise and rationalise the basic principles behind the design of modern computer systems; recognise the features and differences between main architectural families;[9]
3. Analyse and evaluate the impact of architectural design choices on system performance; the importance of memory organisation and caching on machine performance;[10]
4. Develop basic skills for computer architecture design; produce code in assembler and a hardware description language;[12]
5. Develop an analytical approach to evaluating computer systems and competing commercial architectures;[14]

Key/transferable skills

6. Gain experience describing technical design using specialist vocabulary; [17]
7. Plan, develop, evaluate and report on individual pieces of work.[21]

Contents

Introduction to basic CPU architecture. Approaches to CPU design. Amdahl's Law. Example architectures (e.g. ARM, iA64). Assembler, calling conventions, register usage, datapath and control.
Example Hardware Description Languages (e.g. VHDL) and/or assembler language. Pipelining: implementation of a pipeline, hazards, bypasses, exception, assessing the performance.
Memory interface: alignment, endian, cache organisation, virtual and physical addressing, coherence.
Performance enhancements: Pipelining, cache memory, RISC vs CISC architectures, superscalar architectures, multi-threaded and trace-based architectures. Input/output design and implementation

Methods of Learning, Teaching and Assessment

Total student effort for this module is 100 hours on average.

Learning and Teaching

1. 12, 2h lectures, informing learning outcomes 1-3.
2. 12, 1h workshops/labs, informing learning outcomes 4 -7.
3. 64 h private study (approx), informing learning outcomes 1-7.

Assessment

1. Two individual assignments (simulation, analysis and programming assignments) that show the student's ability to define and confine a problem domain, model solutions and implement them. This method carrying 40% of the total mark to assess learning outcomes 4-5.
2. One 2-hour unseen final written examination to assess students' knowledge of the theoretical aspects of computer architecture and design; as well as students' assembler and hardware description coding skills. This method carries 60% of the total mark and assesses learning outcomes 1-5.

Feedback given to students in response to assessed work

Feedback will be provided for each assessed component in written form as appropriate. Any coursework will be returned to students with feedback on the accompanying coursework turn in sheet. If students require additional feedback, they are welcome to speak with the TA, and the module-laeder.

Developmental feedback generated through teaching activities

Dialogue between students and staff in workshops and Labs.

Reading List

Stallings, W., "*Computer Organisation and Architecture*", 8th Edition, Prentice Hall, ISBN: 0130493074, (2010).

M. Morris Mano, "*Computer System Architecture*", 3rd Edition, Prentice Hall International Editions, (1993).