# SECR2033 COMPUTER ORGANIZATION & ARCHITECTURE 2019/2020-2

**ASSIGNMENT 1 (10%)**

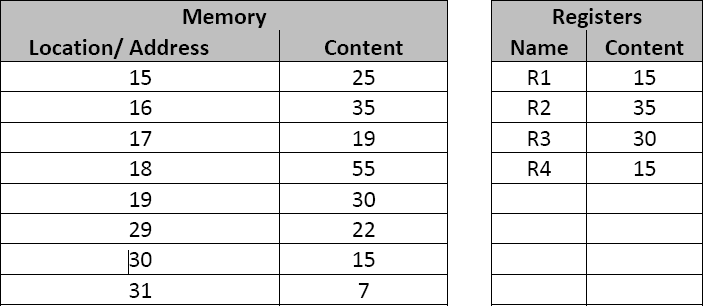
**Date : Friday, 1st May 2020 by 9 AM Duration to complete : 48 hours**

**Deadline and submission: Sunday, 3rd May 2020 by and before 9AM through E-learning**

|  |  |  |
| --- | --- | --- |
| **NAME** | **:** | HAM JING YI |
| **MATRIC NUMBER** | **:** | A19EC0048 |
| **SECTION** | **:** | 02 |
| **LECTURER** | **:** | Dr. Zuriahati |

**Task 1** (7 marks)

1. Consider the following content of memory and registers:



Given that A = 29, B = 3 and C = 17. Determine the value of each operand given the addressing modes in the following table:

# Table 1

|  |  |  |
| --- | --- | --- |
| **Operand** | **Addressing Mode** | **Value** |
| 30 | Immediate | 30 |
| A | Direct | 22 |
| C | Indirect | 30 |
| R2 | Register | 35 |
| R3 | Register Indirect | 15 |
| R1, B | Displacement Addressing | 55 |

1. Match each instructions with the addressing modes given in Table 2.

# Table 2

|  |  |
| --- | --- |
| **Label** | **Addressing Modes** |
| A | Immediate |
| B | Direct |
| C | Indirect |
| D | Register |
| E | Register Indirect |
| F | Displacement |

**Table 3**

|  |  |
| --- | --- |
| **Instruction** | **Addressing Modes** |
| **SUB EDX,(AX)** | E. Register Indirect |
| **MOV AL,AH** | D. Register |
| **ADD EAX,04004000H** | A. Immediate |
| **MOV EBX,(SCORE)** | C. Indirect |
| **MOV EBP,(ARRAY1 + 4)** | C. Indirect |
| **MOV ESI,SUM2** | B. Direct |
| **ADD ECX,(EBX)** | E. Register Indirect |
| **SUB ECX,10110B** | A. Immediate |

**Task 2** (31 Marks)

Your task is to analyze the performance of four different imaginary design of instruction set architectures (ISA). The design of imaginary ISAs are described as

* 1. *Stylo A*—
     + Uses of one-address instruction format.
     + All operations occur between a single register named accumulator and a memory location.
  2. *Stylo B*—
     + Uses of three-address instruction format
     + All three operands of each instruction are in memory.
  3. *Stylo C—*
     + Uses of two-address instruction format.
     + All operations occur between any single general-purpose register and a memory location.
     + There are 8 general-purpose registers, and register specifiers are 3 bits long.
  4. *Stylo D* —
     + Uses of zero-address instruction format.
     + All operations occur on top of the stack.
     + Only push and pop operations access memory; all other instructions remove their operands from stack and replace them with the result.

To measure the performance, you are required to select memory efficiency criteria (i.e., refer to number of bytes generated by each instruction line by line), and make the following assumptions about all the above four instruction sets:

* The opcode is 1 byte (8 bits).
* All memory addresses are 4 bytes (32 bits).
* All data operands are 8 bytes (64 bits).
* All instructions are an integral number of bytes in length.

Invent your own assembly language mnemonics and write the best equivalent assembly language code for the high-level-language code fragment given below and assume the variables A, B, C, D and E are initially in memory:

A = B + C; B = A + C; D = A − B;

E = (B÷A)× D;

1. Write the four code sequences for each ISA stylo of the above code fragments using your invented assembly language.

|  |  |  |  |
| --- | --- | --- | --- |
| Stylo A: 1 address | Stylo B: 3 address | Stylo C: 2 address | Stylo D: 0 address |
| ; A = B + C  LOAD B  ADD C  STOR A  ;B = A + C  ADD C  STOR B  ;D = A − B  LOAD A  SUB B  STOR D  ;E = (B÷A)× D  LOAD B  DIV A  MUL D  STOR E | ; A = B + C  ADD A, B, C  ;B = A + C  ADD B, A, C  ;D = A − B  SUB D, A, B  ;E = (B÷A)× D  DIV B, B, A  MUL E, B, D | ; A = B + C  MOV R1, B  ADD R1, C  MOV A, R1  ;B = A + C  ADD R1, C  MOV B, R1  ;D = A − B  MOV R2, A  SUB R2, B  MOV D, R2  ;E = (B÷A)×D DIV R1, A  MUL R1, D  MOV E, R1 | ; A = B + C  PUSH B  PUSH C  ADD  POP A  ;B = A + C  PUSH A  PUSH C  ADD  POP B  ;D = A − B  PUSH A  PUSH B  SUB  POP D  ;E = (B÷A)×D  PUSH D  PUSH B  PUSH A  DIV  MUL  POP E |

1. Calculate the instruction bytes fetched and the memory-data bytes transferred.

For 0 address (Stylo D), since there are 13 instructions consist of both opcode and memory addresses and 5 instructions consists of only opcode

Instruction bytes fetched = 13(1+4) + 5(1) = **70 bytes**

For 1 address (Stylo A), since there are only 12 instructions consist both opcode and memory addresses,

Instruction bytes fetched = 12(1+4) = **60 bytes**

For 2 address (Stylo C), since there are 11 instructions consist of 1 opcode, 1 register specifire and 1 memory address,

Instruction bytes fetched = 11(1+ (3/8) + 4) = 59.125 bytes (**59 bytes**)

For 3 address (Stylo B), since there are 5 instructions consists of 1 opcode, and 3 memory addresses,

instruction bytes fetched = 5(1+4+4+4) = **65 bytes**

For 0 address (Stylo D), there are 13 data operands,

memory-data bytes transferred = 13(8) = **104 bytes**

For 1 address (Stylo A),there are 12 data operands

memory-data bytes transferred = 12(8) = **96 bytes**

For 2 address (Stylo C), there are 11 data operands,

memory-data bytes transferred = 11(8) = **88 bytes**

For 3 address (Stylo B), there are 15 data operands,

memory-data bytes transferred = 15(8) = **120 bytes**

1. Which architecture is most efficient as measured by code size?

Stylo B (3 address) as it has minimum instructions.

1. Which architecture is most efficient as measured by total memory bandwidth required (in number of bytes for instruction code + data)?

Stylo C (2 address) (59 + 88 = 147 bytes) as stylo C has the least total memory bandwidth.

**Task 3** (12 marks)

Why ARM processor is used in smartphone processor apart from Intel x86 processor?

Write a short essay to answer the above argument. Be detailed and concrete in your analysis that might cover issues, concerns, limitations, constraints, ISA design, etc.

|  |
| --- |
| ANS:  ARM processor is used in smartphone processor apart from Intel x86 processor because x86 processor which is Complex Instruction Set Computer (CISC) takes many cycles to execute an instruction. When there are many instructions, it will consume much time. It also consists of many instructions. This will need a larger CPU to execute, consume much power and run slower. Intel x86 is old enough until the computer engineers don’t know how to design it better. In the other hand. ARM is Reduced Instruction Set computer (RISC) is only a small set of instructions (typically 32). Moreover, there are only simple instructions for the executions. Each instruction is executed in one clock cycle. RISC is effective in using pipelining. By comparing of the CISC and RISC, we can say that CISC can perform complex tasks by executing less instructions and it will take more time, whereas RISC will do the same job by executing more instructions, and it will take less time. RISC is more simple to design, lower energy consumption and cheaper.  In my knowledge, there are only 3 companies have the correct licenses to build x86 compatible processors. This means that only few people can modify or design the x86 based mobile chips. On the other side, ARM has CPU cores under the brand of cortex A and GPUs under brand of Mali. The Cortex-A processor is designed for complex compute task, hosting operating system, supporting the software applications and embedded designs. Whereas the x86 processor spend a lot of power, that means use extra transistors to execute some speculative instructions and branch prediction to avoid unnecessary stalls in the CPU data path. ARM Cortex is architecture design that empowers mobile processors. The ARM which has the RISC base allowed ARM processors require much less transistors than the traditional processors. The benefits of this approach is lower cost, less heat consumption and fewer power usage. There are a lot of processors produced through these two lines. ARM licensed its design to other semiconductor manufactures. So, there will be a lot of ARM-based mobile chips are made and it is convenient for the companies to directly take these design and make use of them.  ARM chips are more suitable for the networking infrastructure applications. The infrastructure need the flexibility, small size, efficiency and low price of the ARM processors. The high-performance storage solutions, servers and routers need the ARM processors due to the advantages above. As we know, the smartphone is getting better and better and it must be able to support a high performance, so it need ARM processor which have 64 bit bits CPU architecture. ARM keeps on delivering high performance and high efficiency by increasing the smartphone’s CPU performance by 100x and GPU by 300x since 2009. ARM, which has fixed instruction formats use only simple decoding logic, waste of memory space and limited of addressing modes. Whereas x86 processor is difficult to decode (use sequential decoding), use compact machine codes but it can accommodate versatile addressing modes.  In conclusion, x86 is mainly used in the field of PC while ARM is used in mobile field because the power consumption of the x86 is very high whereas ARM has a low power consumption. Besides, x86 generates more heat and it requires active heat dissipation while ARM generates less heat and the active heat dissipation is not required. The smartphone using the x86 processor has a lacking battery life compared to the ARM processor. So, based on the explanation above, ARM processor is used in smartphone processor apart from Intel x86 processor. |