Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Roll Number \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

Computer Organisation and Architecture (15B11CI313/10B11CI613)

B.Tech 3rd Semester (CSE&IT)

T3 examination, December 2015

Time: 2 Hours Max. Marks: 70

Weightage: 35%

1. Calculate the effective CPI for an architecture based on the following measurements of average CPI for various types of instructions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction type** | **%age of occurrence** | **Clock cycles** | |
| ALU instructions | 45% | 1.0 | |
| Load-stores | 38% | 1.4 | |
| Conditional Branches | 15% | Taken | 2.0 |
| Not taken | 1.5 |
| Jumps | 2% | 1.2 | |

Assume that 60% of the conditional branches are taken. [6]

1. A DMA module is transferring characters to memory using transparent mode (cycle stealing) from a device transferring at 4800 bps. The processor is fetching instructions at a rate of 106 instructions per second. Determine the maximum IO data transfer rate for DMA. [6]
2. A 5-stage pipelined processor has Instruction Fetch (IF), Instruction Decode (ID), Operand Fetch (OF), Execute (EX) and Write Back (WB) stages. The IF, ID, OF and WB stages take 2 clock cycle each for any instruction. The EX stage takes 1 clock cycle for ADD, 2 clock cycles for SUB instructions, 3 clock cycles for MUL instruction, and 4 clock cycles for DIV instruction respectively. Operand forwarding is used in the pipeline. What is the number of clock cycles needed to execute the following sequence of instructions? Show complete working. [13]

|  |  |
| --- | --- |
| **Instruction** | **Meaning of instruction** |
| *I*0 :MUL R2 ,R0 ,R1 | R2 ← R0 \* R1 |
| *I*1 :ADD R5 ,R3 ,R4 | R5 ← R3 + R4 |
| *I*2 : DIV R2 ,R5 ,R2 | R2 ← R5 / R2 |
| *I*3 :SUB R5 ,R2 ,R5 | R5 ← R2 – R5 |

1. Suppose a new instruction is introduced in the MIPS Instruction Set Architecture. The instruction is written as follows: LW Rd, offset(R1+R2).

It loads a word into the register Rd from an address obtained by adding the two specified registers and the given offset. Suggest an instruction format for it, assuming that all opcodes are 6 bits, and instruction length is 32 bits. Give explanation for your answer. [6]

[8]

You may make suitable assumptions about the values in processor registers before a

Solutions

1. Total clock cycles for 100 instructions in the given mix = 45x1.0 + 38x1.4 + 15x(0.6x2.0 + 0.4x1.5) + 2x1.2 = 45 + 53.2 + 18.0 + 9.0 + 2.4 = 127.6

Effective CPI = 127.6/100 = 1.276 cycles *(For any calculation mistake but otherwise correct reasoning, deduct 2 marks)*

1. DMA module transfer rate is 4800 bps as given. Processor is fetching instruction at 1 MIPS.

Assumption: Processor frequency is 1MHz, so there are 106 processor clock cycles/sec.

DMA is working via cycle stealing. So whenever it fetches data 1 byte it has to steal the processor cycle to have access to the data bus.

1 Byte of data transfer in DMA takes 8 clock cycles.

BR (Bus Request) and BG (Bus Grant) takes 4 clock cycles.

Total clock cycles required for 8 bit data transfer = 12

Total Number of time DMA may steal processor cycle is 106/12

Maximum data transfer that can occur is= 8\*106/12 bps=.67 \*106 bps

1. Operand Forwarding is done from EX stage to OF stage.

IF, ID, OF and WB takes 2 clock cycles each

ADD takes 1 clock cycle

SUB takes 2 clock cycles

MUL takes 3 clock cycles

DIV takes 4 clock cycles

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instruction** | **Clock Cycle Time** | | | | | | | | | | | | | | | | | | | | | |  |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** |
| **MUL R2, R0, R1** | IF | IF | ID | ID | OF | OF | EX | EX | EX | WB | WB |  |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R5, R3, R4** |  |  | IF | IF | ID | ID | OF | OF |  | EX |  | WB | WB |  |  |  |  |  |  |  |  |  |  |
| **DIV R2, R5, R2** |  |  |  |  | IF | IF | ID | ID |  |  |  | OF | OF | EX | EX | EX | EX | WB | WB |  |  |  |  |
| **SUB R5, R2, R5** |  |  |  |  |  |  | IF | IF | ID | ID |  |  |  |  |  |  |  | OF | OF | EX | EX | WB | WB |

Total number of clock cycles needed is equal to 23. If operand forwarding is done from EX to EX stage then the task may be completed in 21 cycles as below. *Marks division for the four instructions in the given order is 2+3+4+4 = 13*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instruction** | **Clock Cycle Time** | | | | | | | | | | | | | | | | | | | | | |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** |
| **MUL R2, R0, R1** | IF | IF | ID | ID | OF | OF | EX | EX | EX | WB | WB |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R5, R3, R4** |  |  | IF | IF | ID | ID | OF | OF |  | EX |  | WB | WB |  |  |  |  |  |  |  |  |  |
| **DIV R2, R5, R2** |  |  |  |  | IF | IF | ID | ID |  |  |  | OF | OF | EX | EX | EX | EX | WB | WB |  |  |  |
| **SUB R5, R2, R5** |  |  |  |  |  |  | IF | IF | ID | ID |  |  |  | OF | OF |  |  | EX | EX | WB | WB |  |

1. The instruction will have to specify three registers – one as the destination (Rd) and two more for address calculation. Each register will need 5 bits for its address because there are 32 registers in MIPS architecture – a total of 15 bits. Opcode requires 6 bits. Hence the number of bits available for offset will be 32 – 6 -15 = 11 bits. Thus the instruction format is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Opcode | Rd | Rs1 | Rs2 | offset |
| 6 bits | 5 bits | 5 bits | 5 bits | 11 bits |

*Explanation for the answer is a must. No explanation but otherwise correct format should be awarded at most 3 marks.*

1. The instruction code has 8 hex digits meaning 32 bits or 4 bytes. The first 2 bytes contain the opcode and addressing mode information. The next two bytes contain the displacement value. Thus from the instruction code 01936645h, the displacement D = 4566 (as per Little-endian storage). *Deduct 2 marks for not considering little-endian aspect.*

Hence EA of the required location = BP + DI + D = 078C + 37B4+ 4566 = 84A6

PA of required location = DS \* 16 + EA = 452A0 + 84A6 = 4D746

*If displacement not taken into account, and otherwise correct, award only 4 marks.*

1. Assumptions: (a) when the function sumArray is called, Register E is having the n value, and the address p is available in register pair HL. (b) Other registers are important and hence will be saved on stack in the beginning.

*2 marks for making apprpriate assumptions*

sumArray: PUSH PSW

MVI C, 00H ;for accumulating the sum in C

MVI B, 00h; to take care of sum exceeding 8-bits

next: MOV A, M ; data in acc. x = \*ptr (*2 marks upto this)*

CPI 05 ; compare x to 5

JC next

JZ next ; considering the case of greater than only *(4 marks for checking the comparison correctly)*

ADD C

JNC save

INR B

save: MOV C, A *(another 3 marks for adding)*

INX H : move to next address

DCR E ; decrease the counter

JNZ next *(3 marks for this preparation for next iteration)*

POP PSW

RET *(1 marks for writing RET)*

The final sum value is available to the calling function in register pair BC.

*As in any programming problem, different students may come up with different versions. Please take care of variations appropriately. Writing PUSH and POP is not essential*

1. (i) Control word: D7 = 1 (for I/O mode), D6-D5 = 01 (Port A in mode 1), D4 = 0 (port A as output, D3 = 0 (don’t care), D2 = 1 (port B in mode 1), D1 = 1 (port B as input), D0 = 0 (don’t care).

Hence control word = 1010 0110 = A6 *1 mark for finding correct control word*

Program MVI A, A6

OUT 0F; send control word to control register

*Another 1 mark for the two lines of correct code*

(ii) Port B is to be used in interrupt driven I/O mode. Hence INTEB should be enabled which is through PC2 bit (status word). It is done through BSR mode. *1 mark for correctly identifying the bit PC2 from the given status word format*

Control word: D7 = 0 (BSR mode), D6-D4 = 000 (don’t care), D3-D1 = 010 (selecting PC2), D0 = 1 (bit set)

Hence control word = 0000 0101 = 05 *1 mark for finding correct control word*

Program MVI A, 05

OUT 0F; send control word to control register

*Another 1 mark for the two lines of correct code*

(iii) Program:

LDA 3000h; get the byte from 3000h

MOV C, A ; save it in register C

readStatus: IN 0E; read port C to check status of A port

ANI 80; 80 is the mask for checking the status of OBFA bar

JZ readStatus; keep checking status till OBFA bar is high indicating readiness of the peripheral device

MOV A, C; get the saved byte in accumulator

OUT 0C; send the byte to port A

*Notes: I have given the explanation in the form of comments. A student may give it in the form of running text or as comments, but explanation is necessary. Correct program without explanation should be awarded at most 3 marks. For program with explanation, the break up can be :*

*Getting the byte from memory 1 mark*

*Status check loop 3 marks*

*Sending the byte to port A 1 mark*

(b part) memory mapped I/O

1. Instead of OUT command we shall use STA with 16-bit address of control register
2. Instead of OUT command we shall use STA with 16-bit address of control register
3. Instead of IN command in the status check loop we shall use LDA command with 16-bit address of port C; and for OUT command, we shall use STA command with 16-bit address of port A

*Each part has 2 marks. There is no need to give actual port addresses. The student must mention that now it will be a 16-bit address.*