PES University, Bangalore

(established under Karnataka Act No. 16 of 2013)

In Semester Assessment - 2 (ISA-2) B. Tech. 3rd Semester, Nov. 2020

UE19CS201: Digital Design and Computer Organization

Time: 80 minutes **All questions to be answered** **Max. marks: 40**

1. (a) (4)

(i) Consider the simple shift–add logic used to multiply two numbers studied in class. If the numbers are n-bits in size, how many clock cycles are required to perform the multiplication?

Solution: n clock cycles.

(ii) Consider multiplier logic used to multiply two numbers, the multiplier size being 8-bits. If the multiplier (in binary) is 11001110, how many addition operations would be required if the simple shift–add multiplier is used? How many addition/subtraction operation would be required if the Booth multiplier is used?

Solution: 5 and 3.

(b) (4)

(i) Suppose a Wallace tree multiplier is used to multiply two five-bit numbers using the approach discussed in class (only full adders are used, no half adders used or any other optimizations done). The above circuit contains only the carry save adders and not the final stage adder. If the delay of a single full adder is 100 ps, how long does the above circuit take to compute?

Solution: Since the Wallace tree will have three levels of CSAs, 300ps.

(ii) Consider the use of carry save addition technique to add three n-bit numbers, the final addition being performed by ripple carry addition. If the delay of a single full adder (or carry save adder) is t_{FA} then what is the time required for above addition?

Solution: $(n+1)t_{FA}$

(c) (2)

(i) What is the advantage of non-restoring division over restoring division?

Solution: Avoid the need for restoring the current remainder (in accumulator A) after an unsuccessful subtraction (A<0).

(ii) Consider the IEEE 754 single precision floating point format. What are the values of exponent and fraction for representation of infinity? What are the values of exponent and fraction for representation of NaN (Not a Number)?

> **Solution:** Infinity: 11111111 and 0
> NaN (Not a Number): 11111111 and nonzero

2. (a) Consider the MIPS architecture R-type instructions. Draw a diagram of the R-type machine language format, clearly labeling the bit-fields with their names and the number of bits they occupy. (2)

(b) (4)

(i) Consider the instruction `lw $s2, -1($s3)`. Given that its opcode, in binary, is 100011, write down the 32-bit machine language instruction in hexadecimal. *Hint*: Register address encoding for register `$s0` is 10000.

> **Solution:** 100011 10011 10010 1111 1111 1111 1111
> 8E72FFFF

(ii) The branch target address of a J-type instruction needs to be 32 bits in size not all of which are specified in the instruction. How are the remaining bits determined?

> **Solution:** Bits 27 to 2 are specified as a part of the instruction.
>
> (a) Bits 0 and 1 are always zero.
>
> (b) Bits 31 to 28 are copied from the four MSB of PC+4.

(c) Consider the machine language instruction (in hexadecimal) 1231FFFF. Given that 000100 is the opcode for the `beq` instruction, write the assembly language instruction corresponding to the above machine language instruction. What, if anything, can be said about when the branch is taken? What happens when the instruction is executed? (4)

> **Solution:** `beq $s1, $s1, -1`
> Since same register is compared, the branch is always taken. Since the offset is -1, the instruction branches to itself resulting in an infinite loop.

3. (a) Consider a 16-bit microprocessor whose register file has 32 registers. Its register type instructions specify only two registers. For example, "add r1, r2" will add the contents of r1 and r2, and place the results back in r1. So only two registers need to be specified. How many bits are left over to specify the instruction opcode? Assume that all bits are used to specify just the opcode and the two register addresses. (3)

> **Solution:** Five bits required to store encode each of the two registers, consuming ten bits out of the 16-bit word. Therefore six bits are available for opcodes.

(b) Consider the `sw` (load word) MIPS instruction. How many operands does it have and what is the type of each operand? Consider the multicycle MIPS datapath. How many clock cycles does it take to execute the `sw` instruction on the above datapath? Briefly describe what actions happen in each clock cycle. No need to describe control signals but mention what happens in the datapath. No need to draw any diagram. (7)

Solution: The `sw` instruction takes three operands two of which are registers (5-bits each) and one (immediate operand) is an address offset (16-bits).

On the MIPS multicycle datapath, the `sw` instruction takes four clock cycles to execute.

1. Instruction is fetched from memory and stored in IR.
2. The register containing the address is read out using one of the read ports of the register file and stored in the register at the output of the register file. Also, the register whose contents have to be written to memory is also read out using the other read port of the register file and stored in the register at the output of the register file. Meanwhile the address offset is sign extended.
3. The ALU is used to add the register contents to the sign extended offset and the result is stored in the register at the output of the ALU.
4. The above register's contents are used as the write address to the memory and the register contents read out of the other read port are applied as data input to the memory whose write enable signal is also asserted, thereby performing the memroy write and completing the instruction execution.

4. (a) (4)

(i) The MIPS architecture instructions use various addressing modes. Consider the `and` and `j` instructions. For each above instruction specify the addressing mode it uses.

Solution: Register and pseudo-direct addressing respectively.

(ii) Consider the components of a standard computer (PC) presented in class. What gets place on top of microprocessor? Also, add-on cards such as a graphics card attach to the PC via slots on the motherboard. What is the name of the interface provided by the slots?

Solution:

- Heatsink and/or fan
- PCIe or PCI Express

(b) Consider the 3×3 systolic array used to multiply two 3×3 matrices. The systolic array took seven clock cycles to perform the 3×3 matrix multiplication. How many clock cycles would be required to multiply to 32×32 matrices using a 32×32 systolic array? (3)

Solution: 94 clock cycles.

(c) Consider the multiplication of two unsigned n bit numbers. The product is clearly (atmost) a $2n$ bit number. Now consider the multiplication of two two's complement n bit numbers yielding a two's complement product. How many bits (atmost) is the size of the product? Briefly explain why. Also, suppose the product is guaranteed to be a negative number. Then how many bits (atmost) would the size of the product be? Briefly explain why. (3)

Solution: Most positive product $(-2^{n-1})(-2^{n-1})$ would require $2n$ bits. Most negative product $-(2^{n-1})(2^{n-1}-1)$ would require $2n-1$ bits.