UNIVERSITY OF MORATUWA
Faculty of Engineering
Department of Electronic & Telecommunication Engineering
B.Sc. in Engineering
Batch 20 - Process Design Project
EN2031 - Fundamentals of Computer Organization and Design


Time Allowed: 1 week

27th February 2023 midnight

---

**INSTRUCTIONS TO CANDIDATES**:

- This design project accounts for 30% of the module assessment. The total maximum mark attainable is 100. The marks assigned for each question & sections there of are indicated in square brackets.
- ***Pay attention to the marks indicated for each part of the question as they indicate the approximate time that needs to be spent for each part.***
- The answer needs to be type-written. Diagrams can be hand-sketched if necessary and scanned to be included in the answer.
- ***All submitted answers will be sent through plagiarism software to detect similarities and any submission with more than 50% similarty index will be marked zero.***
- If you have any doubt as to the interpretation of the wording of a question, make your own decision, and clearly state it.
- ***Answers must be given in point form where appropriate. Marks will be deducted for unnecessarily descriptive answers.***

1. A custom processor is to be designed for a computational unit and it will be part of a System on Chip (SoC) to be placed in an embedded system for a high-speed Industry 4.0 compliant factory operation. The decision on a custom processor was made following an in-depth investigation which reveals that off-the-shelf embedded processors do not meet the throughput requirements.

   The processor is not expected to run an operating system. A custom compiler will convert C language program to assembly which will be loaded to program memory. Pipeline processing is required to achieve the required throughput.

   The following specifications of the processor has been given to you.

   - Data Memory and Program Memory to be of size 64KB (high speed low latency) and the memory is dual port. Both Program and Data Memory can handle 16-bit words.
   - A registry file **R** contains 8 registers : R0 to R7 where R7 acts as a Stack Pointer.
   - Program Counter (PC) is considered as a special register.
   - SREG is a special circular buffer that holds 8 1-bit values corresponding to results of evaluated condition in branch instructions.
   - 16-bit Arithmetic and Logic Unit (ALU)
   - Dedicated adders are to be used for address incrementing/decrementing where appropriate.
   - Data Memory has an area reserved for a stack (limited to 4KB) and Data Memory is accessed by a separate 16-bit address bus.
   - Load instructions have different variants as given below under Instruction Set details.
   - The instruction set supports the following:
     - Arithmetic instructions on the ALU. The supported operations are addition, subtraction, bitwise AND and bitwise OR.
     - Any register in the registry file can be selected either as an operand or a result register. The selection of the one operand register and the result register must be identical. This is referred to as *read-modify-write constraint*.
     - POP from and PUSH to the Stack. SP is decremented/incremented as part of POP/PUSH instruction using an ALU.
     - Indirect register load from Data Memory. Every register in the Registry file can be specified as an address register and as a destination register. [Hint: If Data Memory is DM, Rd=DM[Ra] where Rd and Ra are any of the registers in Register File **R** except R7 – Stack Pointer.
     - Indirect register load from Data Memory with *address post-modification*. Every register in the Registry file can be specified as an address register and as a destination register. In parallel with the load, the address register is incremented on the ALU (*read-modify-write constraint*).
     - Instruction set needs to facilitate saving a 16-bit value to any register in **R**.
     - Data transfer between any two registers in **R**.
     - Unconditional and conditional branch instructions. The target address is specified relative to the current program counter value. It is supplied as an 8-bit signed immediate value.
     - Conditional branch instructions do register comparisons for equality, less than or equal and greater than or equal conditions. The evaluated conditions are stored in SREG. [*Hint: Consider using one register in the Registry file **R** designated as one register for branch condition comparison purposes*]
     - 8-bit load immediate instruction. An 8-bit signed constant is loaded to any register. 8-bit immediate value is signed extended to a 16-bit value.

   You may assume reasonable values for any information that is not specifically provided above and must clearly state these assumptions. You are expected to propose appropriate opcodes for various instructions.

   This is an open ended design project and marks will be given based on the approach you take in the design process. Wherever required provide appropriate justifications for your decisions.

   You are required to do the following referring to the specifications given above.

(a) i. (20 points) List all the required instructions with proper format showing opcode and the operands and the meaning of each instruction.

ii. (20 points) Develop the Instruction Formats required to support the above-mentioned instructions and provide justifications for your decisions. *(Hint: You need separate formats for ALU, Control Flow, Load, Store and Register move Instructions)*

(b) (15 points) Propose an appropriate encoding scheme for the opcodes that will faciliate easy decoding of the instructions.

(c) (20 points) Draw a clearly labelled datapath with all functional elements for your chosen design approach. Show all the control signals for the datapath elements.

(d) i. (5 points) Explain your chosen design approach for the microarchitecture: Hardwired or Microprogramme with clear justification.

ii. (20 points) Design the controller for the datapath that you have designed above, giving reasons for your decisions. Clearly illustate the controller showing the inputs and outputs.