1

**Project Description:**

This class project is to design and implement a simple processor circuit, which can perform **addition**, **subtraction**, and **unsigned multiplication** operations on registers. Moreover, the processor can perform **three logical operations** between two registers, and **move/branch instructions**. A sequence of assembly instructions (according to the processor assembly language) must be used to test/verify the functionality of all processor units and instruction executions. Use Vivado simulator to show functional simulation results of executing the assembly program.

**Required Specifications:**

You will design and implement a processor using Verilog Hardware Description Language (Verilog HDL) targeting Nexys-4 FPGA board. The following required specifications must be considered in your control unit-datapath design and report.

**Instructions:** the processor must be able to execute fixed-length instructions ofeither one-byte or two-byte instructions. One-byte instructions must have the following format (shown below) to perform **arithmetic** or **logical** operations between two registers. Multiplication must be performed between two registers, and the product must be stored into two registers {RHi:RLo} as 16-bit number. Two-byte instructions must has fields for opcode, destination register address (RD) or source register address (RS), and operand value (address or data), the two bytes must be stored at consecutive memory locations.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 4 | | 3 | 2 | | 1 | 0 | |
|  |  |  |  |  |  |  |  |  |
| Opcode |  |  |  | RS/RD | |  | RS | |
|  | |  |  | |  |  |  |  |
| Figure 1: One-byte instruction format | | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |
| Opcode |  |  | 0 | 0 |  |  | RS/RD |  |
|  |  |  |  |  |  |  |  |  |

Operand

Figure 2: Two-byte instruction format

**Registers:** the processor datapath must have four 8-bit registers: **R0**, **R1**, **R2**,and **R3**. Intuitively, the register addresses will be 00, 01, 10, and 11, respectively. RD and RS could be any of these four registers. Moreover, Program counter (**PC**), **RHi** and **RLo** registers which are to hold the upper byte and lower byte of the product,respectively, and Instruction register (**IR**) must be considered in your design.

**Read Only Memory (ROM):** A 256X8 ROM must be initialized to store theassembly program which will be used to verify the processor operations. The assembly program (shown below) must be converted first into machine codes according to table 1.

Each instruction line (of the assembly program) must be saved in one or two bytes of the ROM starting at address **00**.

**NOP**

**MOV R0, 0**

**MOV R1, 12**

**MOV R2, 5**

**MOV R3, 34**

**ADD R1, R2**

**MUL R1, R2**

**OR R2, R3**

**MOV [25], R2**

**MOV R0, [25]**

**SUB R0, R2**

**JNZ 15**

**JMP 0**

**Table 1: Processor instructions set**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | Opcode | Register |  | Operation |  |
|  | Operation |  |
| Binary |  |
| Mnemonics | Transfer | Type |  |
|  |  |
|  | bits 7-4 |  |  |  |  |
| **NOP** | 0000 | No Operation | No operation | 1-byte |  |
|  |  |  |  |  |  |
| **ADD** RD, RS | 0001 | RDRD+RS | Add register to register | 1-byte |  |
| **SUB** RD, RS | 0010 | RDRD-RS | Subtract register to register | 1-byte |  |
| **MUL** RS1, RS2 | 1100 | {RHi:RLo} RS1\*RS2 | Multiply register to register | 1-byte |  |
| **NOT** RD, RS | 0011 | RD  RS**’** | Bitwise logical NOT | 1-byte |  |
| **AND** RD, RS | 0100 | RDRD&RS | Bitwise logical AND | 1-byte |  |
| **OR** RD, RS | 0101 | RDRD|RS | Bitwise logical OR | 1-byte |  |
| **MOV** RD, RS | 0110 | RDRS | Move register | 1-byte |  |
| **MOV** [Operand], R | 0111 | M[Operand]  R | Store register to memory | 2-byte |  |
| **MOV** [Operand],RHi2**[1]** | 1110 | M[Operand] RHi, | Store product to memory | 2-byte |  |
| M[Operand+1} RLo |  |
|  |  |  |  |  |
| **MOV** R, Operand | 1000 | R  Operand | Load register immediate | 2-byte |  |
| **MOV** R, [Operand] | 1001 | RD  M[Operand] | Load register from memory | 2-byte |  |
| **JMP** Operand | 1010 | PC  Operand | Unconditional jump | 2-byte |  |
| **JNZ** Operand | 1011 | if R0 0, PCOperand | Jump if not zero | 2-byte |  |
|  |  |  |  |  |  |

**[1] As a two-byte instruction, RS/RD field is 00 for this instruction.**

**Random Access Memory (RAM):** A 256X8 RAM must hold data to betransferred from/into registers. Usually processors have instruction memory and data memory.

**Arithmetic and Logic Unit (ALU):** ALU must process 8-bit binary numbersaccording to operations described in table 1. The ALU must accept two unsigned binary numbers (8 bit such as 10011100) and selection lines logic as inputs. Depending on the select lines logic, the circuit will perform the requested unsigned arithmetic/logic operation of the two given binary numbers.