**LAB 5 PART 5 - EXTENDED ISA**

INSTRUCTION ENCODINGS

|  |  |  |  |
| --- | --- | --- | --- |
| OP-CODE  (bits 31-24) | RD / IMM  (bits 23-16) | RT  (bits 15-8) | RS / IMM  (bits7-0) |

mult 4 1 2 (multiply value in register 1 by value in register 2, and place the result in register 4)

sll 4 1 0x02 (apply logical shift left 2 times on value in register 1, and place the result in register 4)

srl 4 1 0x02 (apply logical shift right 2 times on value in register 1, and place the result in register 4)

sra 4 1 0x02 (apply arithmetic shift right 2 times on value in register 1, and place the result in register 4)

ror 4 1 0x02 (apply rotate right 2 times on value in register 1, and place the result in register 4)

bne 0x02 1 2 (if values in registers 1 and 2 are not equal, branch 2 instructions forward)

ASSIGNED OPCODES

op\_mult = 8’b0000\_1000;

op\_sll = 8’b0000\_1001;

op\_sla = 8’b0000\_1010;

op\_srl = 8’b0000\_1011;

op\_sra = 8’b0000\_1100;

op\_ror = 8’b0000\_1101;

op\_bne = 8’b0000\_1110;

ALU FUNCTIONS

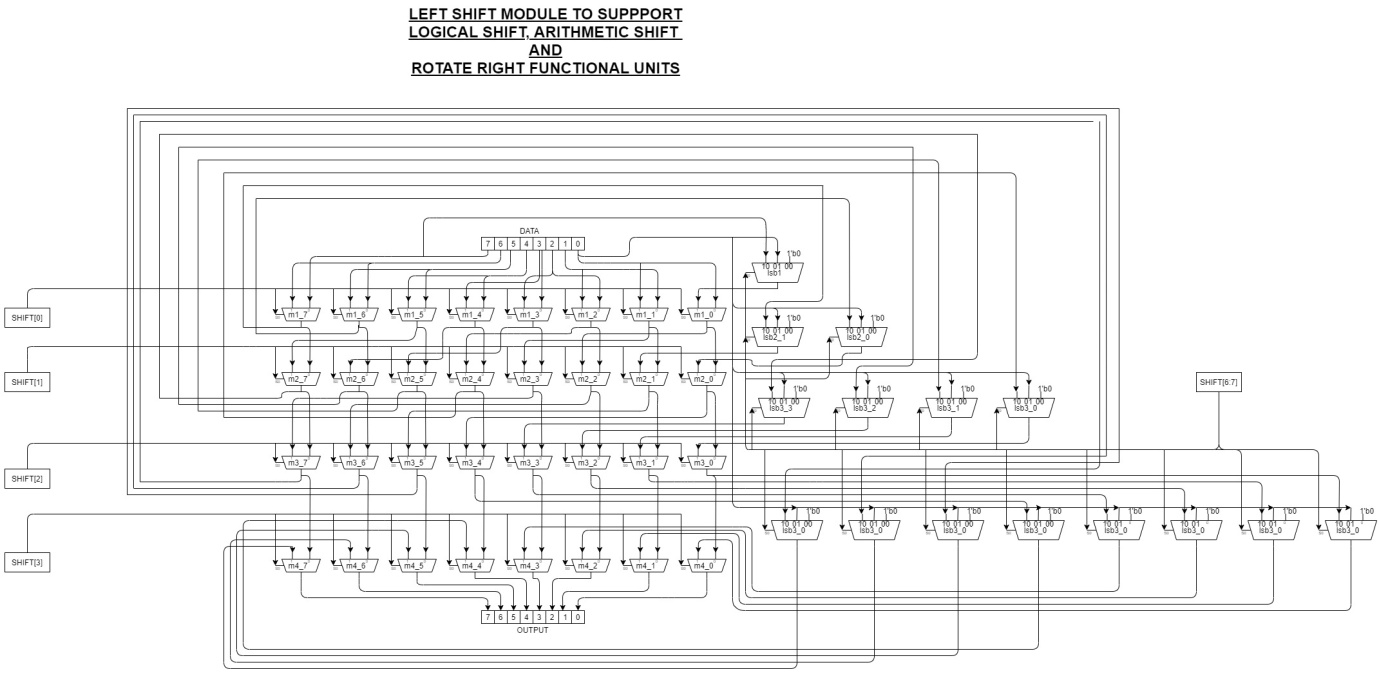
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SELECT** | **Function** | **Description** | **Supported**  **Instruction** | **Unit’s**  **Delay** |
| 100 | MULTIPLIER | DATA1\*DATA2 -> RESULT | mult | #3 |
| 101 | LEFT SHIFT  ROTATE RIGHT | DATA1<<DATA2 -> RESULT DATA1<<<DATA2 -> RESULT  DATA1(ROTATE)DATA2 -> RESULT | sll, sla, ror | #3 |
| 110 | RIGHT SHIFT | DATA1>>DATA2 -> RESULT  DATA1>>>DATA2 -> RESULT | srl, sra | #3 |

SIGN BIT CONVENTIONS FOR SHIFT

sll = “00” sla = “01” ror = “10”

srl = “00” sra = “10”

LEFT SHIFT FUNCTIONAL UNIT



Module supports : Logical Left Shift, Arithmetic Left Shift, Rotate Right

For shifts greater than 8bits, maximum shift of 8bits is done.

RIGHT SHIFT FUNCTIONAL UNIT

Similar module which supports logical right and arithmetic right.

TIMING DELAYS FOR THE DATAPATH

mult :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PC Update | Instruction Memory Read | | Register Read | | ALU |
| #1 | #2 | | #2 | | #3 |
|  | PC+4 Adder |  | Decode |  |  |
| #1 | #1 |
| Register  Write |  | |  | |  |
| #1 |

sl :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PC Update | Instruction Memory Read | | Register Read | | ALU |
| #1 | #2 | | #2 | | #3 |
|  | PC+4 Adder |  | Decode |  |  |
| #1 | #1 |
| Register  Write |  | |  |  |  |
| #1 |

sr :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PC Update | Instruction Memory Read | | Register Read | | ALU |
| #1 | #2 | | #2 | | #3 |
|  | PC+4 Adder |  | Decode |  |  |
| #1 | #1 |
| Register  Write |  | |  |  |  |
| #1 |

bne :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PC Update | Instruction Memory Read | | Register Read | | 2’s Comp | ALU |
| #1 | #2 | | #2 | | #1 | #2 |
|  | PC+4 Adder |  | Branch/Jump Target Adder | |  | |
| #1 | #2 | |
|  | | | Decode |  |  | |
| #1 |