**Milestone 1: R-Type Instructions**

**10% of Project Grade**

**Due: Thursday, November 2**

**Requirements:**

* Design and implement all components shown in the following diagram.
* Design experiments to test your design and analyze your experimental data.

****

**Figure 1. An Example of Five Stage Pipelined Processor**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instruction | OPCode | RS | RT | RD | Shamt | Funct |
| **add** $3, $2, $1 | 000000 | 00010 | 00001 | 00011 | 00000 | 100000 |
| **sub** $3, $2, $1 | 000000 | 00010 | 00001 | 00011 | 00000 | 100010 |
| **and** $3, $2, $1 | 000000 | 00010 | 00001 | 00011 | 00000 | 100100 |
| **or** $3, $2, $1 | 000000 | 00010 | 00001 | 00011 | 00000 | 100101 |

**Grading Criteria:**

1. **(70 points)** Design three test programs to verify the correctness of the basic diagram without forwarding. Show your assembly code, binary code, and how the verification is processed.
2. **(15 points)** Design two test programs to verify the correctness of forwarding from EX/MEM to ALU. Show your assembly code, binary code, and how the verification is processed.
3. **(15 points)** Design two test programs to verify the correctness of forwarding from MEM/WB to ALU. Show your assembly code, binary code, and how the verification is processed.

**Milestone 1: R-Type Instructions (10% of Project Grade)**

**TA Checkoff Sheet**

Names: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Score: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TA Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Initialize the register *i* with a value of *i*, where *i* = 0, 1, …, 31. For example, register 11 is initialized as 0x0000000A. Verify the correctness of your design via the VGA display.

Grading:

1. **(70 points)** Verify the correctness of the basic diagram without forwarding.

|  |  |
| --- | --- |
| Program | Correct? |
| add $20, $1, $2 # $20 = 0x3  sub $21, $5, $3 # $21 = 0x2  add $22, $7, $4 # $22 = 0xB  sub $23, $10, $2 # $23 = 0x8  sub $24, $0, $1 # $24 = 0xFFFFFFFF  and $25, $7, $6 # $25 = 0x6  or $26, $14, $9 # $26 = 0xF  and $27, $8, $7 # $27 = 0x0  or $28, $16, $11 # $28 = 0x1B | $20 =  $21 =  $22 =  $23 =  $24 =  $25 =  $26 =  $27 =  $28 = |

1. **(15 points)** Verify the correctness of forwarding from EX/MEM to ALU.

|  |  |
| --- | --- |
| Program | Correctness |
| add $20, $1, $2 # $20 = 0x3  sub $20, $20, $3 # $20 = 0x0  add $22, $7, $4 # $22 = 0xB  sub $22, $10, $22 # $22 = 0xFFFFFFFF (-1)  sub $24, $0, $1 # $24 = 0xFFFFFFFF (-1)  add $24, $24, $24 # $24 = 0xFFFFFFFE (-2) | $20 =  $20 =  $22 =  $22 =  $24 =  $24 = |

1. **(15 points)** Verify the correctness of forwarding from MEM/WB to ALU.

|  |  |
| --- | --- |
| Program | Correctness |
| add $20, $1, $2 # $20 = 0x3  sub $1, $0, $3 # $1 = 0xFFFFFFFD  add $2, $20, $4 # $2 = 0x7  sub $1, $5, $1 # $1 = 0x8  add $2, $2, $2 # $2 = 0xE  sub $24, $0, $1 # $24 = 0xFFFFFFF8  add $24, $24, $24 # $24 = 0xFFFFFFF0  add $24, $24, $24 # $24 = 0xFFFFFFE0  add $24, $24, $24 # $24 = 0xFFFFFFC0  add $24, $24, $24 # $24 = 0xFFFFFF80  add $24, $24, $24 # $24 = 0xFFFFFF00  add $24, $24, $24 # $24 = 0xFFFFFE00 | $20 =  $1 =  $2 =  $1 =  $2 =  $24 =  $24 =  $24 =  $24 =  $24 =  $24 =  $24 = |