VE370 Hw2 周振纳 5180219110 39

1. addi $t0, $0, 15
   sll $t0, $t0, 24
   lb $s1, 2($t0)
   The content of $s1 is 0x00000047

2.
```
FACT: addi $sp, $sp, 8      <= addi $sp, $sp, -8
      sw $ra, 4($sp)
      sw $a0, 0($sp)
      add $s0, $0, $a0
      slti $t0, $a0, 2      <= slti $t0, $a0, 1
      beq $t0, $0, L1
      mul $v0, $s0, $v0     <= addi $v0, $0, 1
      addi $sp, $sp, -8     <= addi $sp, $sp, 8
      jr $ra
L1:   addi $a0, $a0, -1
      jal FACT
      addi $v0, $0, 1       delete this line
      lw $a0, 4($sp)
      lw $ra, 0($sp)
      addi $sp, $sp, 8      <= { addi $sp, $sp, 8
                                 mul $v0, $a0, $v0
      jr $ra
```

3. 1).

| 0000 | 0001 | 0110 | 1010 | 0100 | 1000 | 0010 | 0010 |
|---|---|---|---|---|---|---|---|
| 0 | 11 | 10 | 9 | 0 | 22 | | |

   sub $t1, $t3, $t2

   2). R-type

4. 1).

| op | $s2 | $s1 | -32 |
|---|---|---|---|

| 100011 | 10010 | 10001 | 1111111111100000 |
|---|---|---|---|

   Therefore, the binary code is 1000 1110 0101 0001 1111 1111 1110 0000

   2). I-type

5. 1). $128 = 2^7$

| op | rs | rt | rd | shamt | funct |
|---|---|---|---|---|---|
| 6 bit | 7bit | 7bit | 7bit | 5bit | 6 bit |

Total = 38 bits

2).

| op | rs | rt | constant or address |
|---|---|---|---|
| 6 bit | 7bit | 7bit | 16 bit |

Total = 36 bits

6. 1). $64 = 2^6$

| op | rs | rt | constant or address |
|---|---|---|---|
| 6 bit | 6bit | 6 bit | 14 bit |

For beq, new PC = current PC + 4 + Relative address × 4.

Since the bits for relative address is changed from 16 to 14.

Therefore, the range of address for beq is reduced.

2).

| op | rs | rt | rd | shamt | funct |
|---|---|---|---|---|---|
| 6bit | 6bit | 6bit | 6bit | | 6bit |

For jr, it is R-type and PC = R[rs].

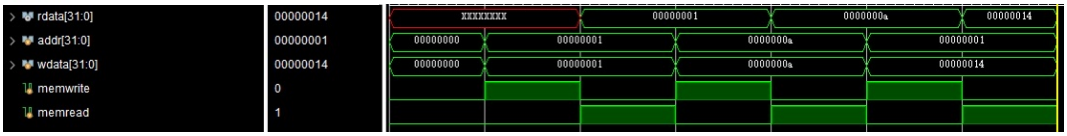
Therefore, the range of address for jr is unchanged.

7.

| | | | | | | |
|---|---|---|---|---|---|---|
| 0x1000F400 | 000000 | 00000 | 01000 | 01101 | 00000 | 101010 |
| 0x1000F404 | 000101 | 01101 | 00000 | 0000000000000001 | | |
| 0x1000F408 | 000010 | 0000 0000 0000 1111 0100 0001 10 | | | | |
| 0x1000F40C | 001000 | 10011 | 10011 | 0000000000000010 | | |
| 0x1000F410 | 001001 | 01010 | 01010 | 0000000000000001 | | |
| 0x1000F414 | 000010 | 0000 0000 0000 1111 0100 0000 00 | | | | |
| 0x1000F418 | | | | | | |

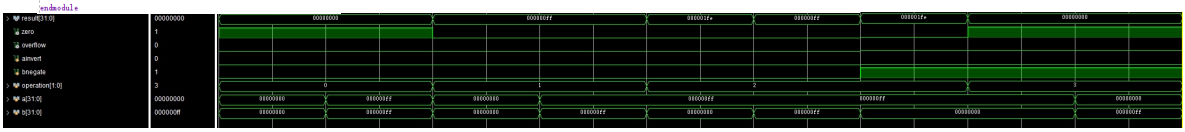
8.

```
module datamemory (rdata, addr, wdata, memwrite, memread);
    input [31:0] addr, wdata;
    input memwrite, memread;
    output [31:0] rdata;
    reg [31:0] rdata;
    reg [31:0] mem[1023:0];
    always @ (addr or wdata or memwrite or memread)
    begin
        if (memwrite == 1) mem[addr] = wdata;
        else if (memread == 1) rdata = mem[addr];
    end
endmodule
```




9.

```
module ALU(ainvert, bnegate, operation, a, b, result, zero, overflow);
    input [31:0] a, b;
    input ainvert, bnegate;
    input [1:0] operation;
    output [31:0] result;
    output zero, overflow;
    reg [32:0] c;
    wire [32:0] d;
    wire t;
    reg [31:0] result, ta, tb;
    always @ (ainvert or a)
    begin
        if (ainvert == 1) ta = ~a;
        else ta = a;
    end
    always @ (bnegate or b)
    begin
        if(bnegate == 1) tb = ~b + 1;
        else tb = b;
    end
    always @ (ainvert or bnegate or operation or a or b)
    begin
        case (operation)
        2'b00: result[31:0] = ta & tb;
        2'b01: result[31:0] = ta | tb;
        2'b10: result[31:0] = ta + tb;
        2'b11:
        begin
            assign c = ta + tb;
            if (c[31] == 0) result[31:0] = 32'b0;
            else result[31:0] = 32'b1;
        end
        endcase
    end
    assign d[32:0] = ta + tb;
    assign overflow = d[32];
    assign t = (result[31:0] == 32'b0);
    assign zero = t;
endmodule
```



10.

| Executable File Header | | |
|---|---|---|
| | Text size | 0x440 |
| | Data size | 0x90 |
| Text Segment | Address | Instruction |
| | 0x00400000 | lui $at, 0x1000 |
| | 0x00400004 | ori $a0, $at, 0x0000 |
| | ... | ... |
| | 0x00400084 | jr $ra |
| | ... | ... |
| | 0x00400140 | sw $a0, 0x8020($gp) |
| | 0x00400144 | jmp 0x004002C0 |
| | ... | ... |
| | 0x004002C0 | jal 0x00400000 |
| | ... | ... |
| Data Segment | Address | |
| | 0x10000000 | (X) |
| | ... | ... |
| | 0x10000020 | (Y) |
| | ... | ... |