|  |  |  |
| --- | --- | --- |
| CSc 4210/6210 | **COMPUTER ARCHITECTURE** | Spring 2019 |

 Assignment 3 Due by: 11:59 PM, March 3, 2019.

|  |  |  |  |
| --- | --- | --- | --- |
| First name |  | Last name |  |

1. The block diagram shown below represents the hardware that implements the following microoperations:

(ab`c`+ abc`)L: R1 ← R2`

(ab`c+abc)L: R1 ← R1 + R2

S

R1

0

Complementer

L

R2

1

Adder

Where R1and R2 are two n-bit registers and a, b, and c are control variables. Include the logic gates for the control functions and multiplexer inputs and outputs. (Remember that the symbol + designates an OR operation in a control or Boolean function but it represents an arithmetic plus in a microoperation.)

Simplify the control statements for the complementer, adder and the load input (L);

1. Complementer= (ab`c`+ abc`)L. The simplified complementer expression= (ab'c'+abc') = ac'(b'+b) = ac' (since b+b' = 1)
2. Adder=(ab`c+abc)L. The simplified adder expression=  ac(b'+b) = ac
3. L= (ab`c`+ abc` + ab`c+ abc). The simplified expression for L = ab'(c+c') + ab(c+c') = ab'+ab = a(b'+b) = a
4. The selelction multiplexer switch is S=  A'ac' + Aac
5. Three registers; A, B and C have 16 bits each where A holds the following hexadecimal value; 4F74, and B holds the following octal value; 145771.
6. Determine the binary output C for C=(A**.**B)L, C=(A+B)L, C=(A+B), C=(A+B`+1), C=A⊕B, C= ashl B, and C=ashr B (single shift only) microoperations. Negative values are in 2’s complement.
7. Indicate the cases where an over-flow is detected.
8. The given diagram is for the connections between the three registers, the ALU unit and a multiplexer. Indicate the size of the multiplexer, the selection switches, and the number of inputs

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reg | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | operation |
| A | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | Store A |
| B | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | Store B |
| C | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | (A.B)L |
| C | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | (A+B)L |
| C | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | (A+B`+1)L |
| C | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | (A⊕B)L |
| C | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | ashl B |
| C | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | ashr B |

1. Overflow bit

MUX

8x1

s0 s1 s2

so

ALU

|  |  |
| --- | --- |
| operation | Overflow bit |
| (A.B)L | 0 |
| (A+B)L | 1 |
| (A+B`+1)L | 0 |
| (A⊕B)L | 0 |
| ashl B | 0  B  C  A |
| ashr B | 0 |

S0

S

1. A multiplexer is as bus to connect 8 registers, R0, R1,R2,R3,R4,R5,R6,R7

The load input of each register is controlled by a decoder.

1. Complete the missing labels of the circuit diagram
2. Complete the missing information in the following table.

DEC D0

D1

D2

Load D3

D4

D5

0 D6

1

2 D7

D

MUX

0

1

2

3

4 y

5

6

7

A0

A1

A2

s0

s

s

A

|  |
| --- |
| R0 |
| R1 |
| R2 |
| R3 |
| R4 |
| R5 |
| R6 |
| **R7** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A2 | A1 | A0 | S2 | S1 | S0 | Register name (destination) | Register name (source) |
| 0 | 0 | 0 | 1 | 1 | 1 | R0 | R7 |
| 0 | 0 | 1 | 1 | 1 | 0 | R1 | R6 |
|  |  |  |  |  |  | R2 | R5 |
| 0 | 1 | 1 | 1 | 0 | 0 | R3 | R4 |
| 1 | 0 | 0 | 0 | 1 | 1 | R4 | R3 |
| 1 | 0 | 1 | 0 | 1 | 0 | R5 | R2 |
| 1 | 1 | 0 | 0 | 0 | 1 | R6 | R1 |
| 1 | 1 | 1 | 0 | 0 | 0 | R7 | R0 |

1. Write an RTL expressions that satisfy the following conditional statement

If ( A = 1) then (R0 R1)

else if ( B = 1 ) then ( R0 R2)

else if ( C = 1 ) then ( R0 R3 )

1. Insert the load (L) condition to the left of the colon for each statement.

(A==1) :( R0 R1)

(A==0&&B==1) : (R0 R2)

(!A&&!B&&C) :(R0 R3)

1. Complete the following table;

Use the selection s1s0 as follows; (00) tor R0, (01) for R1, (10) for R2 and (11) for R1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | L | S1 | S0 |
| 0 | 0 | 0 | Nothing loaded | 0 | 0 |
| 0 | 0 | 1 | (!A&&!B&&C) | 1 | 1 |
| 0 | 1 | 0 | (!A&&B) | 1 | 0 |
| 0 | 1 | 1 | (!A&&B) | 1 | 0 |
| 1 | 0 | 0 | (A==1) | 0 | 1 |
| 1 | 0 | 1 | (A==1) | 0 | 1 |
| 1 | 1 | 0 | (A==1) | 0 | 1 |
| 1 | 1 | 1 | (A==1) | 0 | 1 |

1. Simplify S1 and S0 in terms of A, B and C.

S0=A+B`C

S1= A`B+A`C