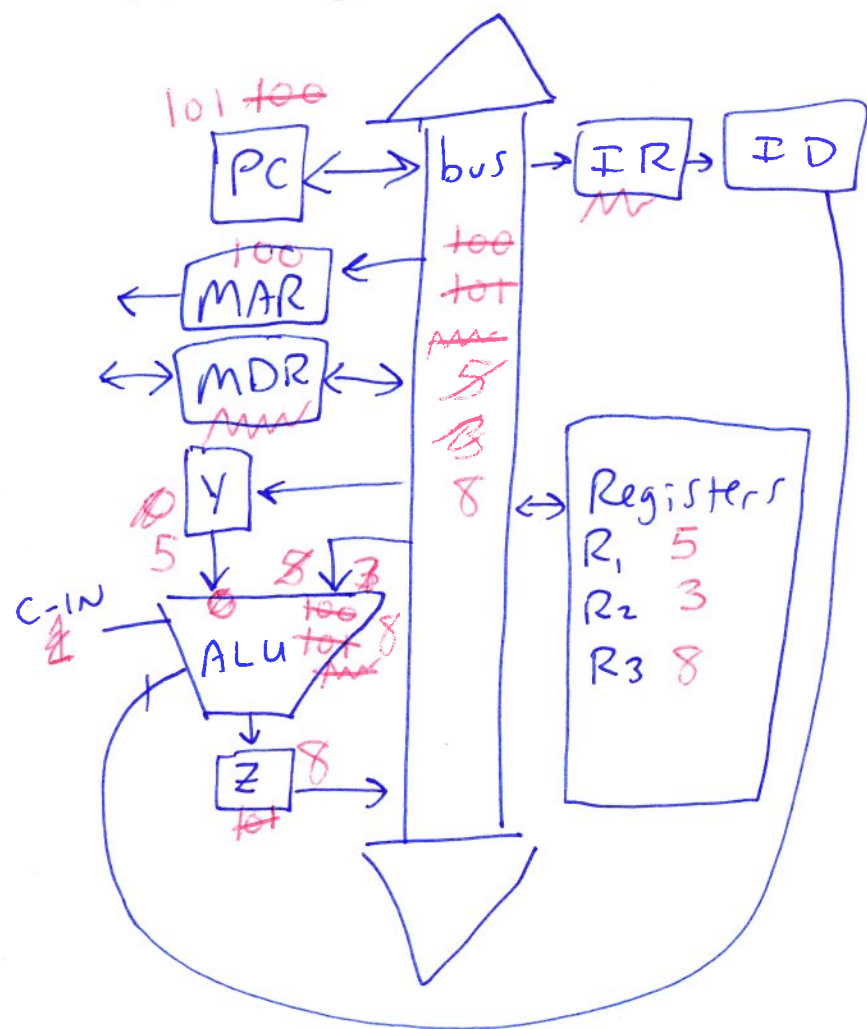


2721

fourth lesson after midterm

-1-

Microcode to F, D, E "Add R1, R2, R3"
 which means add the operand in register 1 to the
 operand in R2, and store the sum in R3, as
 efficiently as possible.



PC_{OUT}, MAR_{IN}, READ,
 1. CLR Y, Set C-IN, ADD, Z_{IN}

2. Z_{OUT}, PC_{IN}, WMFC

3. MDR_{OUT}, IR_{IN}, DECODE

"Add R1, R2, R3"

4. R1_{OUT}, Y_{IN}

5. R2_{OUT}, ADD, Z_{IN}

6. Z_{OUT}, R3_{IN}, END.

1
 88
 +88
 176

Write a microprogram to F, D, E

-2-

"Add R1, R2, R2, R3"
↖ just stores the sum

1. PCout, MARin, READ, CLR Y, Set Cin, ADD, Zin
2. Zout, ~~PCin, Decode~~ PCin, WMFC
3. MDRout, IRin, Decode
4. ~~MDRout~~ R2out, Yin, ADD, Zin
5. Zout, Yin
6. R1out, ADD, Zin
7. Zout, R3in, END.

Write a microprogram to execute the instruction:

-3-

Add (R1), R2, R3 stores the sum.
R1[↑] stores the mm
address of the operand

4. R1_{out}, MAR_{in}, READ, ~~WAF~~

5. R2_{out}, Y_{in}, WMFC

6. MDR_{out}, ADD, Z_{in}

7. Z_{out}, R3_{in}, END.

Execute ^{ADD} (R4), (R5), R6

-4-

4. R4_{out}, MAR_{in}, READ, WMFC

5. MDR_{out}, Y_{in}

6. RS_{out}, MAR_{in}, READ, WMFC

7. MDR_{out}, ADD, Z_{in}

8. Z_{out}, R6_{in}, END.

Execute ADD ~~R1~~ (R1), (R2)

-5-

4. R1_{out}, MAR_{in}, READ, WMFC

5. MDR_{out}, Y_{in}, ADD, Z_{in}

6. Z_{out}, MDR_{in}

7. RZ_{out}, MAR_{in}, WRITE, WMFC, END.

ADD R1, ((R2)), R3

4. R2_{out}, MAR_{in}, READ

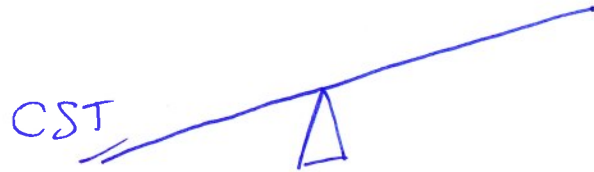
5. R1_{out}, Y_{in}, WMFC

6. MDR_{out}, MAR_{in}, READ, WMFC

7. MDR_{out}, ADD, Z_{in}

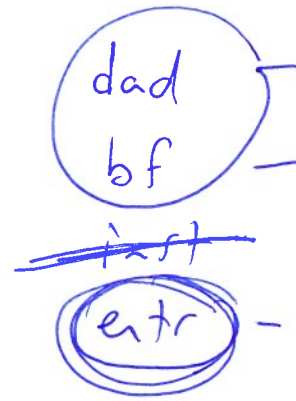
8. Z_{out}, R3_{in}, END.

balance



12:50am

alignment



-7-
-
-7-