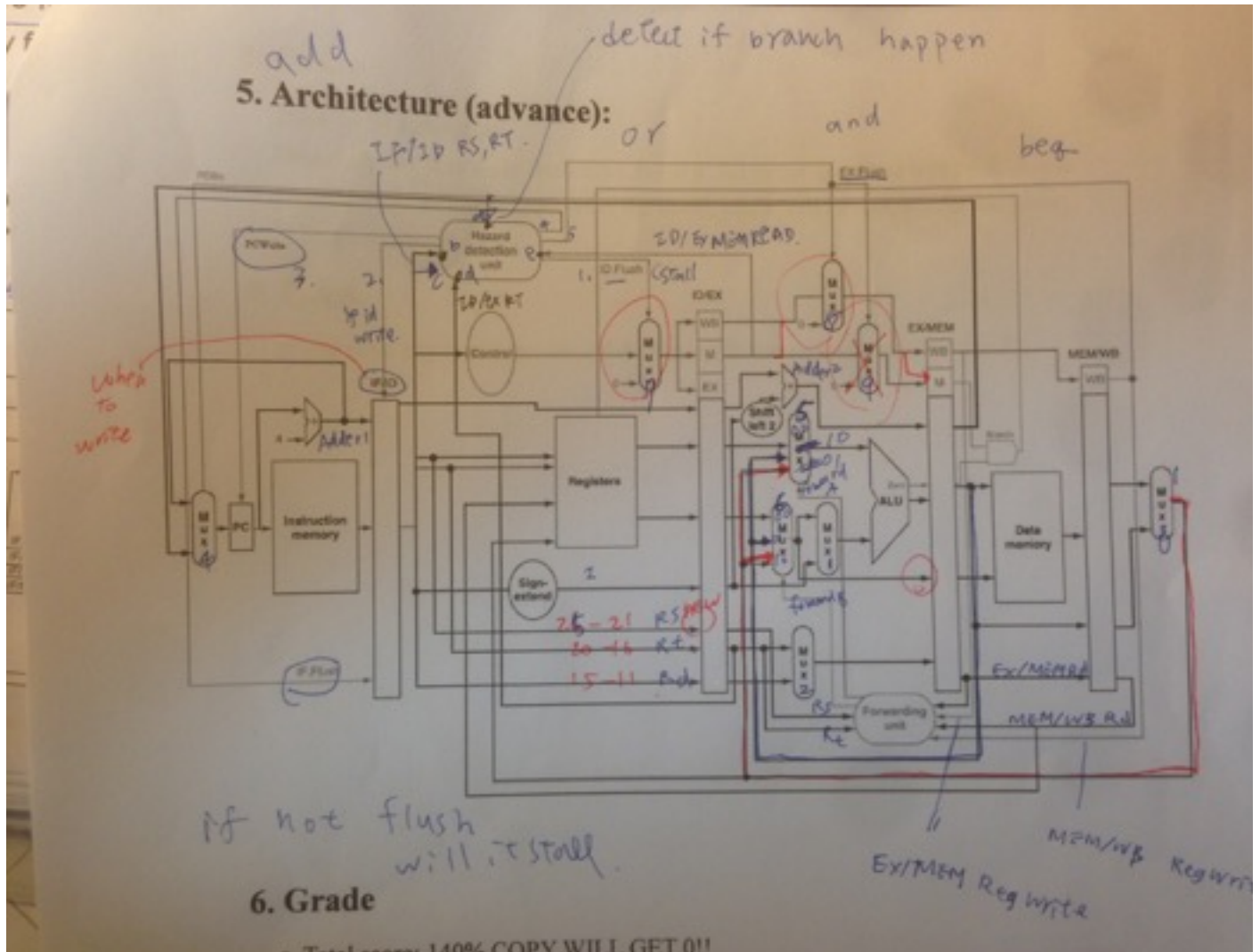


# Computer Organization

Architecture diagram: 2015/06/9版



## Detailed description of the implementation:

這邊都還是只有basic instructions 還沒有做forwarding跟hazard detection。decoder, ALU, ALU control 都沒有改，因為功能都是舊有的。只有ALU pipeline 加了四條 register for temporary storage。

Advanced 的部分，因為hand out給的Architecture 跟講義不同，所以forwardA 跟forwardB的Mux 會調整。

由Hazard Detection unit 出來的 ID flush 跟Ex flush 可以直接連到ID/EX bar 跟 Ex/Mem bar 直接如 IF flush這樣就不用再多做

Mux。但我還是有做來測試兩者的不同。

### Problems encountered and solutions:

利用之前做的版本再加東西就好了，不需要整個都按照上面的圖，像mux3 就只有 輸入的source 不同而已，很好檢查到，只要看圖去檢查，而非一行一行code的checking。

在test3中

```
000000 01010 01011 00010 00000 101010 slt $2, $10, $11
                                                beq stall
```

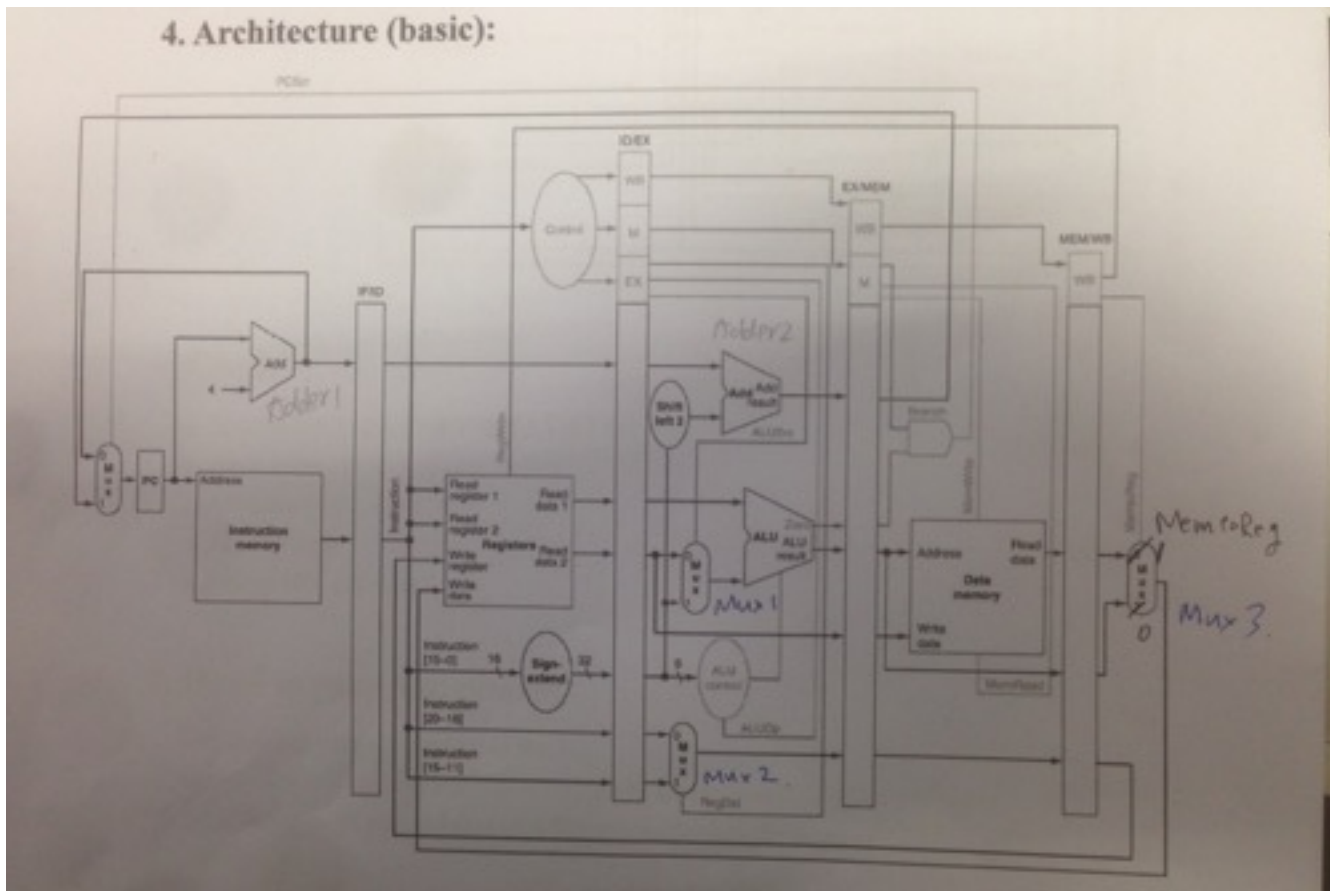
```
000100 00010 00001 00000000000000011 beq$2,$1,off()
```

data hazard for branch

這個要stall。

### Lesson learnt (if any):

成果截圖，測試結果 第一筆test data 都對。在mem stage的branch 其實可以放入除了BEQ其他 Lab3 也有做的branch指令都跟Lab3一樣，只是要多一個mux 我是用4 to 1 mux。 branch type 作為 select 就像是從decoder 傳出來的 branch 一樣，branch 也是從ID stage 經由EX 來到MEM stage。



第二筆test data 也都對

```
Register=====
r0= 0, r1= 3, r2= 4, r3= 1, r4= 6, r5= 2, r6= 7, r7= 1
r8= 1, r9= 0, r10= 3, r11= 0, r12= 0, r13= 0, r14= 0, r15= 0
r16= 0, r17= 0, r18= 0, r19= 0, r20= 0, r21= 0, r22= 0, r23= 0
r24= 0, r25= 0, r26= 0, r27= 0, r28= 0, r29= 0, r30= 0, r31= 0

Memory=====
m0= 0, m1= 3, m2= 0, m3= 0, m4= 0, m5= 0, m6= 0, m7= 0
m8= 0, m9= 0, m10= 0, m11= 0, m12= 0, m13= 0, m14= 0, m15= 0
r16= 0, m17= 0, m18= 0, m19= 0, m20= 0, m21= 0, m22= 0, m23= 0
m24= 0, m25= 0, m26= 0, m27= 0, m28= 0, m29= 0, m30= 0, m31= 0

** VVP Stop(0) **
** Lesson learnt (if any):
** Flushing output streams.
** Current simulation time is 10010000 ticks.
> finish
```

```
WARNING: Instr_Memory.v:40: $readmemb(C0_P4_test_2.txt): Not enough words in the file for the requested range [0:31].
Register=====
r0= 0, r1= 16, r2= 20, r3= 8, r4= 16, r5= 0, r6= 24, r7= 26
r8= 0, r9= 100, r10= 0, r11= 0, r12= 0, r13= 0, r14= 0, r15= 0
r16= 0, r17= 0, r18= 0, r19= 0, r20= 0, r21= 0, r22= 0, r23= 0
r24= 0, r25= 0, r26= 0, r27= 0, r28= 0, r29= 0, r30= 0, r31= 0

Memory=====
m0= 0, m1= 16, m2= 0, m3= 0, m4= 0, m5= 0, m6= 0, m7= 0
m8= 0, m9= 0, m10= 0, m11= 0, m12= 0, m13= 0, m14= 0, m15= 0
r16= 0, m17= 0, m18= 0, m19= 0, m20= 0, m21= 0, m22= 0, m23= 0
m24= 0, m25= 0, m26= 0, m27= 0, m28= 0, m29= 0, m30= 0, m31= 0

** VVP Stop(0) **
```