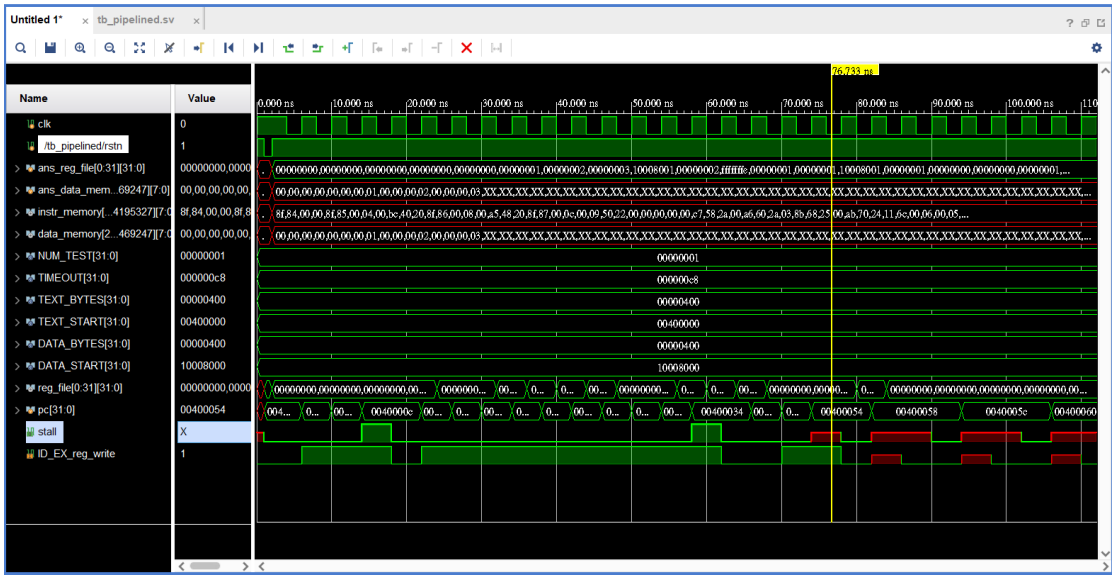
Lab4

111550088 張育維

1. Experimental Result

1. Show the waveform screen shot of the test we provided.



2. What other cases you've tested? Why you choose them?

I test the case that insert the addi $1 $5 24 right before the beq and right after the beq, because the original case doesn't include addi.

2. Answer the following Questions

1. List out the equation to detect EX & MEM hazard in forwarding unit. Which part of the equation in textbook p. is wrong?

```
if(EX_MEM_reg_write && EX_MEM_rd == ID_EX_rs && EX_MEM_rd != 0)
    forward_A = 2'b10;
else if(MEM_WB_reg_write && MEM_WB_rd == ID_EX_rs && MEM_WB_rd != 0)
    forward_A = 2'b01;
else
    forward_A = 2'b00;
if(EX_MEM_reg_write && EX_MEM_rd == ID_EX_rt && EX_MEM_rd != 0)
    forward_B = 2'b10;
else if(MEM_WB_reg_write && MEM_WB_rd == ID_EX_rt && MEM_WB_rd != 0)
    forward_B = 2'b01;
else
    forward_B = 2'b00;
if(EX_MEM_rd == IF_ID_rs && EX_MEM_reg_write)
    forward_branch[1] <= 1'b1;
else forward_branch[1] <= 1'b0;
if(EX_MEM_rd == IF_ID_rt && EX_MEM_reg_write)
    forward_branch[0] <= 1'b1;
else forward_branch[0] <= 1'b0;
```

In P.369 of the textbook, the equations are wrong:

```
if (MEM/WB.RegWrite
and (MEM/WB.RegisterRd ≠ 0)
and not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
       and (EX/MEM.RegisterRd ≠ ID/EX.RegisterRs)
and (MEM/WB.RegisterRd = ID/EX.RegisterRs)) ForwardA = 01

if (MEM/WB.RegWrite
and (MEM/WB.RegisterRd ≠ 0)
and not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
       and (EX/MEM.RegisterRd ≠ ID/EX.RegisterRt)
and (MEM/WB.RegisterRd = ID/EX.RegisterRt)) ForwardB = 01
```

Because there should be more one ')' right after '…ID/EX.RegisterRs' in the line 4 and '…ID/EX.RegisterRt' in the last line 2. I found the correct version in the fifth edition:

```
if (MEM/WB.RegWrite
and (MEM/WB.RegisterRd ≠  0)
and not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠  0)
       and (EX/MEM.RegisterRd ≠  ID/EX.RegisterRs))
and (MEM/WB.RegisterRd = ID/EX.RegisterRs)) ForwardA = 01

if (MEM/WB.RegWrite
and (MEM/WB.RegisterRd ≠  0)
and not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠  0)
       and (EX/MEM.RegisterRd ≠  ID/EX.RegisterRt))
and (MEM/WB.RegisterRd = ID/EX.RegisterRt)) ForwardB = 01
```

2. In forwarding for beq , is forwarding from MEM/WB to ID needed? Why?

No, because if we want the result of MEM/WB part, the instruction must be load instruction. However, when branch reads the register right after load instruction, we will insert two stalls. After two stalls the register has been already written the new value.

3. Briefly explain how you insert stalls when beq reads registers right after lw writes it.

I set that if 'mem_read' in ID/EX or in EX/MEM equals to one, stall would occur. Because in all instructions we have to implement, only lw needs to read the data from the memory. Therefore, I just need to observe whether this instruction read the data from the memory to check whether it is lw. Also, We need two stalls when meet the lw, so I check the EX/EX and EX/MEM registers to

make sure that put the lw in the MEM/WB correctly.

4. sw right after lw is quite common since copy and paste a data from one address to another is used frequently. In textbook, a stall is followed by the lw in this case. Is it possible to remove this stall? How?

Yes, we can resolve this by data forwarding. When lw is in the MEM stage, the sw is in the EX stage. In the next clock cycle, we can use the data that lw got to replace the data that sw read from the register, and store in the correct address.