1.Convert32: (wenzhuo)

12bit -> 32 bits

2.Convert12 (wenzhuo)

32 -> 12

3.brcomp (haoyu)

branch comparator P38,BrEQ改BrNEQ，其他沿用。并且只会是signed

4.CP5WBsel (haoyu)

4:1 mux

5.control logic (jal, setx, bex)

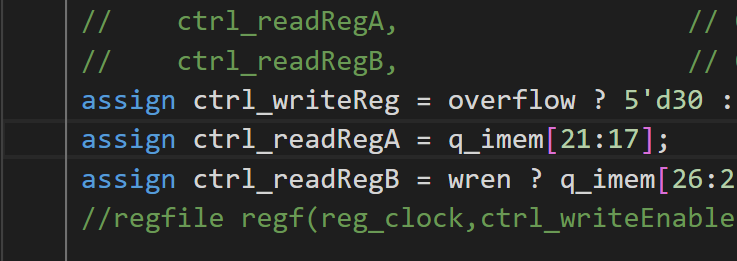
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| insn | code | Immsel | ctrl\_writeEnable | Bsel | ALUSel | wren(MemRW) | WBSel | PCSel | BrNEq | BrLT | Asel |
| add/sub…. | 00000 | 0 | 1 | 0 | Alucode | 0 | 1 | 0 | X | X | 0 |
| ·addi | 00101 | 1 | 1 | 1 | 00000 | 0 | 1 | 0 | X | X | 0 |
| sw | 00111 | 1 | X | 1 | 00000 | 1 | X | 0 | X | X | 0 |
| lw | 01000 | 1 | 1 | 1 | 00000 | 0 | 0 | 0 | X | X | 0 |
| J T | 00001 | 1 | X | 1 | 00000 | 0 | X | 1 | X | X | 0 |
| bne | 00010 | 1 | X | 1 | 00000 | 0 | X | ? | 1->pcsel=1 | X | 1 |
| jal | 00011 | 1 | 1 | 1 | 00000 | 0 | 2 | 1 | X | X | 1 |
| jr | 00100 | X | 0 | 1 | 00000 | 0 | X | 1 | X | X | 0 |
| blt | 00110 | 1 | X | 1 | 00000 | 0 | X | ? | X | 1->pcsel=1 | 1 |
| bex | 10110 | 1 | X | 1 | 00000 | 0 | X | ? | 1->pcsel=1 | X | 0 |
| setx | 10101 | 1 | 1 | 1 | 00000 | 0 | 1 | 0 | x | x | 0 |

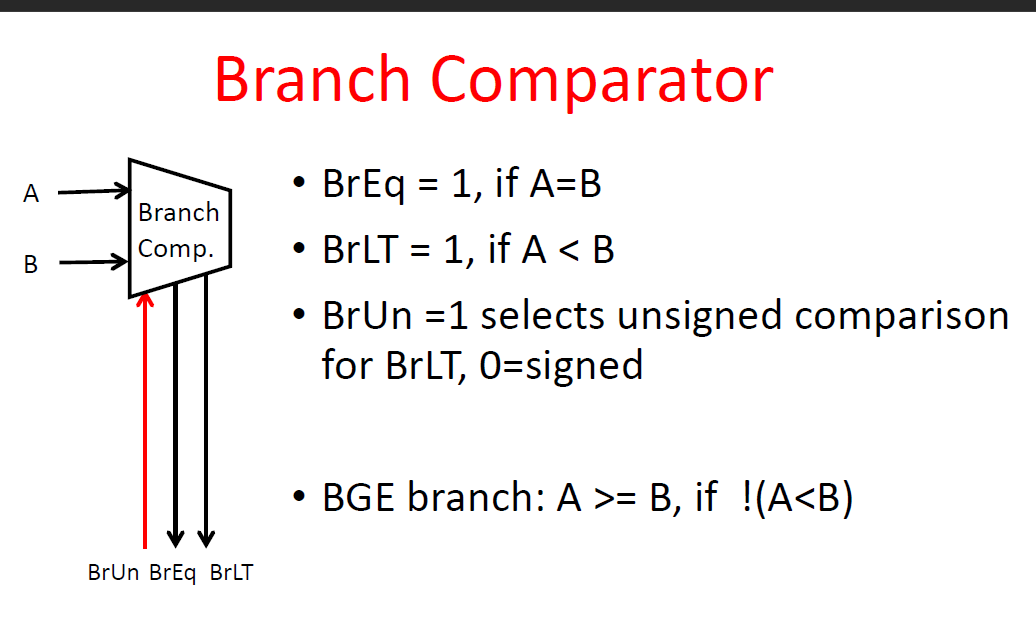
6.

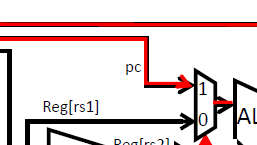
setx -> addi (Addr–D[26:22]=$rstatus)

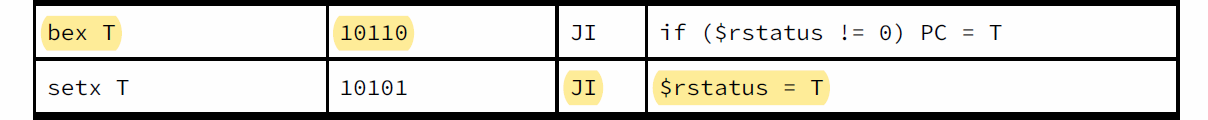
Bex -> bne (Addr–A(B)[26:22]=$rstatus)

Jal -> ctrl\_WriteReg\_destination=$31(134行加一个mux）









改input register值

We designed the modules and interfaces together and assigned the module to each member to implement. I am responsible for signal extension module, which is sx.v, Convert12.v and Convert32.v. Finally we integrated the modules and debug the code using waveform and testbench.

When bne is triggered, the data fed into the branch comparator is compared, and then go to the control logic, at the same time, PC will be added to go the next instruction.

lw instruction. Because when this command is triggered, there are many hit and miss operations in the cache.

1ns

Because there are many modules need different clocks, we need to set different frequency clocks to trigger the instructions. For imem and dmem, the clock should be faster than regfile and PC register because they should be prepared to fed before regfile and PC register. In addition, rising edges and falling edges should not overlap in different clocks to avoid conflict.

The module pc4.v is used to compute the next PC, whenever triggering this module, the address\_imem is added to 1 using an RCA, so as to achieve the operation of adding 1 in address and go to the next PC.

We searched the processor design in some academic papers and designed the schematic by hand. Then we broke down into different modules and implemented separately and integrated to debug.

In the command extending module, the input and output port were reversed. When testing the waveform, the output didn't meet the expectation, so I checked the code logic and the interface order, and found the parameters were in the wrong order. After correcting it, the module worked.