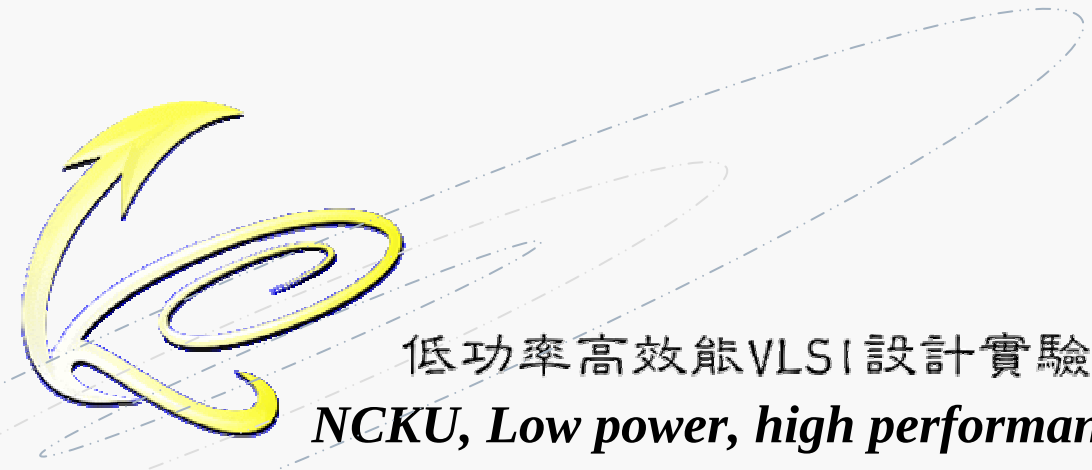
Lab 4
CPU Verification

Fall 2008



低功率高效能VLSI設計實驗室
NCKU, Low power, high performance VLSI design lab

CPU Specification

- The CPU design so far has the following specifications:
  - Implement the 12 instructions as listed
  - Using FSM to design the controller
  - Register File size: 32x32bits
  - Assume Instruction Memory and Data Memory with no delay
  - Instruction memory size: 32x32bit
  - Data memory size: 32x32bits
  - timescale 1ns/10ps
  - Clock period: 10ns

CPU Integration

- Put all major components together along with other small components
- Run SMILE using machine code -- “mins.prog”
- Simulated waveform “.vcd” files
- Report:
  - Block diagram with detail descriptions of each block
  - Define the interfaces between each block clearly
  - Simulation results with explanation
  - Conclusion (for all members)

CPU Verification I: Executing the “ mins.prog

mins.prog content

- 0. LD R0 <=21
- 1. SW M0<=R0
- 2. SW M1<=2'hff
- 3. LD R1<= M1
- 4. LD R0<= M0
- 5. LD R1<=10
- 6. ADD R1=R1+R0;
- 7. SUB R1=R1-R0
- 8. AND R1=R0 & R1
- 9. OR R1=R0 || R1
- 10. LD R0<=7
- 11. XOR R0=R1^R0
- 12. ADD R0=R0+4'b1101
- 13. NOP
- 14. SUB R1=R1-5'b10000
- 15. SLL R0=R0 SLL(R1)
- 16. SRL R0=R0 SRL(R1)
- 17. NOP
- 18. LD R1<=5'b11100
- 19. RLL R0=R0 RLL(R0)
- 20. RRL M0=R0 RRL(3)
- 21. LD R2<= M0
- 22. ADD R2=R2+4'b1010

CPU Verification II

- Write a machine code that could carry out the arithmetic operations and show that the answers are correct.

1. **120 + 360**
2. **1-65536**
3. **255*(-7)**
4. **40*2 - 160/4**

CPU Verification III

- Write a machine code that could carry out the following logic operations and show that the answers are correct.

1. **A=1010_1010, B=1111_0000, C=0000_1111, D=0101_0101, E=((A AND B) OR C) XOR D, find E.**

2. **L = 1010_1010_1010_1010_1010_1010_1010_1010,**
   M = 0101_0101_0101_0101_0101_0101_0101_0101,
   N = 1111_0000_1111_0000_1111_0000_1111_0000,
   X = Shift Left L by 4 bits, follow by Rotate Left by 8 bits
   Y = Shift Right M by 4 bits, follow by Rotate Right by 8 bits
   Z = Shift Left N by 8 bits, follow by Rotate Right by 8 bits