Control design

2024年6月13日 12:59

Instruction decoding logic

1. **Signal input**
   There are two input, op and func of the circuit. The control circuit unit will send signal to the whole cup by the input of op and func. The op and func are all 6-bit.
2. **Signal detector**
   The circuit will first determine the op code. The circuit have different option for the input to select which instruction will execute. Op code is mainly for I-type instruction.
   If the op code is 00, it will access func determine. Use to distinguish different kinds of R-type instruction and send signal to other parts of CPU.
3. **Signal decoding**
   After detected input code, we use a comparator to decoding the instruction to generate different control signal, the signal will dictate how the subsequent circuit handles the current instruction.
4. **I-type instruction generate**
   The green lines in the diagram represent the paths of the control signals. Each control signal corresponds to a specific operation, such as:
   - LW: Load Word instruction
   - SW: Store Word instruction
   - BEC: Branch if Equal instruction
   - BNE: Branch if Not Equal instruction
   - ADDI: Add Immediate instruction
5. **R-type instruction generate**
   The upper right part of the diagram deals with R-Type instructions, such as ADD and SLT.
   These instructions use the FUNC field to further specify the exact operation.
6. **Special instruction**
   SysCall: System call instruction
   These instructions are handled the halt of the counter.

ALU control logic

This part will send signal to ALU to dictate which function will execute.

1. Inputs:
   OP: This is the opcode input, representing the main operation code for the instruction.
   FUNC: This is the function code input, used mainly for R-Type instructions to specify the exact operation.
2. Logic Gates:
   AND Gates:
   There are AND gates connected to the OP and FUNC signals. These gates help in determining specific operations based on the combination of the OP and FUNC values.
   OR Gate:
   The output from the AND gates is fed into an OR gate. This OR gate combines the signals to determine the correct ALU_OP.
3. MUX (Multiplexer):
   The MUX (Multiplexer) selects between different inputs based on the control signal b. It has two inputs:
   Input 0: One of the inputs from the logic gates.

Input 5: Another possible input from the logic gates.
The selection line (b) determines which input is passed through to the output.

4. Output:
ALU_OP: The output of the MUX is the ALU_OP signal. This signal is used to control the operation of the ALU, determining which arithmetic or logic operation to perform.

Detailed Signal Flow:

- The OP input is split and sent to multiple AND gates.
- The FUNC input is also split and combined with the OP inputs in the AND gates.
- The outputs of the AND gates are combined using an OR gate to generate the appropriate signals.
- The MUX then selects the correct signal based on the control input (b).
- The selected signal is output as ALU_OP, which controls the ALU's operation.

This setup allows the control unit to generate the correct ALU_OP signal based on both the main opcode and the function code, enabling the ALU to perform the required operation for the instruction being executed.

Ps. To meet the requirement of the assignment, and to simple the design of CPU, the control just open few function in ALU. Although there are sub, mul, div and other functions, we just use the add function and compare function in ALU. Other functions will individual testing in the final part.

Instruction decoding signal

- **RegDst**:
  It is used to determine whether RT is used as the target register or RD as the target register
- **RegWrite**:
  Determine if Reggie Stefeld needs to write back, R_type, ADDI and LW instruction all need RegWrite signalMemToReg:Determine whether it is necessary to access Memory to take out the data and write it to Register file.
- MemToReg：
  This is used to determine whether data should be taken from memory and bring to register file.
- **MemWrite**：
  This is used to determine whether memory needs to be accessed for writing
- **AluSrc**:
  To be fed into Mux to detect the second into ALU is R2 from register file or immediate numebr from instruction.
- Eeq:
- Used to determine the two number is equal and branch to another instruction.
- Bne
  Used to determine the two is not equal and branch to another instruction.
- Halt:
  Used to determine if the whole circuit should shop.