Computer Science 605.611

Problem Set 5 Answers

1. Suppose that our MIPS core instruction subset (add, sub, slt, and, or, lw, sw, beq & j) is expanded to include another instruction, the or immediate (ori) instruction. Answer each of the following questions about possible modifications to the multi-cycle datapath and FSM that may be required to support this additional instruction.

a) (3) What is the minimum number of new states required for the FSM (finite state machine) description of the ori instruction execution?
Two new states would be required: state 10 in which the OR of the zero-extended immediate with the rs register is performed; and state 11 in which the ALU output is written into the rt register (not the rd register as in state 7). The bits that normally identify the rd register (bits 11 through 15) are part of the immediate operand field for I-type instructions.

b) (3) Identify any new control bits that the control unit has to output for the ori instruction.
The lower B input to the ALU would now have 5 possible candidates: the constant 4, the rt register, the sign-extended immediate, the sign-extended immediate shifted left 2 bits & the zero-extended immediate. So ALUSrcB would have to be expanded from 2 bits to 3 bits (by including one new bit) and a larger MUX is needed to select from among the 5 inputs. If instead the sign-extension unit is modified to also produce a zero-extended immediate, a new control bit is needed to tell the modified extension unit which type of immediate to generate.

c) (3) For each new state required for ori, list all control bit settings used in the new state and indicate what determines when to transition into the new state.
In the new state 10, the 3-bit ALUSrcB should be set to 100 to select the zero-extended immediate as the lower ALU B input, ALUSrcA=1, and ALUOp should be set to the unused pattern 11 to denote an OR operation. State 10 is entered following state 1 based on the ori opcode. State 11 follows state 10. In state 11, RegDst=0, RegWrite=1 & MemtoReg=0 to write the ALU output to the rt register. State 7 can't be used since it sets RegDst=1 which selects rd as the result register.

d) (3) Identify any new hardware devices that are required in the datapath to support the execution of the ori instruction.
A zero-extension unit is required. Alternatively, the existing extension unit could be modified to generate either the signed-extended or the zero-extended immediate. This would require a new control bit to tell the extension unit to sign or zero extend.

2. The micro-programmed implementation of our MIPS multi-cycle datapath employs a micro-PC.
a) (3) How many bits does the micro-PC contain?
The micro-PC requires 4 bits to indicate the state number (0 through 9).
Each state corresponds to a separate micro-instruction.

b) (3) What do the bits within the micro-PC indicate? They indicate which micro-instruction to use next (i.e., the index of the micro-instruction).

c) (3) When the micro-PC is incremented, by how much is it incremented?
Unlike the PC, the micro-PC is incremented by 1.

d) (3) When the micro-PC is updated, what bits within the micro-instruction indicate how to update the micro-PC?
The 2 LSBs address control (AdrCtl) indicate how to update the micro-PC (set it to 0, increment it, or use ROM1 or ROM2 LUT (lookup table) to obtain the next value for the micro-PC).

3. Both of our MIPS single-cycle and multi-cycle datapaths use control bits to indicate the operations and direct the actions that take place when executing machine instructions.
a) (3) How many control bits are required for our MIPS single-cycle datapath?
There are 10 control bits for the single-cycle datapath to support our MIPS core instruction subset.

b) (3) How many control bits are required for our MIPS multi-cycle datapath?
There are 16 control bits required for the multi-cycle datapath.

4. If the control unit for the single-cycle datapath were implemented using a PLA (programmable logic array), what is the minimum number of inputs and the minimum number of outputs required for the PLA? Name each of the inputs and the outputs.
a) (5) minimum inputs are: _4_the 6 opcode bits (op5 through op0) identify the type of instruction. However for the core subset Op0 through Op3 are sufficient to distinguish one instruction from the other in the subset.

b) (5) minimum outputs are: ___there are 10 datapath control bits (ALUOp1, ALUOp0, RegDst. RegWrite, MemtoReg, ALUSrc, MemRd, MemWrt, Branch, Jump) required of our MIPS core instruction subset._______

5. a) (3) Why do CISC processors tend to use micro-programming?
CISC systems tend to use micro-code because it is easier to use micro-programming to implement the more complex instructions.

b) (3) Why do RISC processors not use micro-programming?

RISC systems do not use micro-code because hardwired logic is faster. Multiple micro-instructions must be retrieved and interpreted for each machine instruction while hardwired RISC systems execute the machine instructions directly.

c) (3) What is the main reason that ROM rather than RAM is used to store micro-code (i.e., micro-programs)?
Using ROM avoids the possibility of corrupting the micro-code through accidental or intentional attempts to modify the micro-code. Also the micro-code is retained in ROM even when the power is off. RAM is volatile and loses its contents when the power is off.

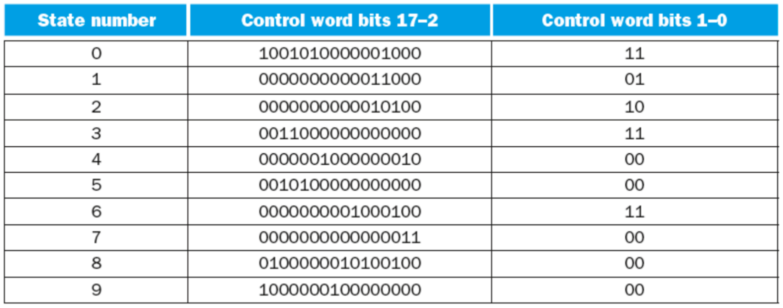
6. Answer each of the following questions about the micro-programmed implementation of our MIPS multi-cycle datapath described in module 5:

a) (3) How many bits are contained within each machine instruction? All machine instructions contain 32 bits.

b) (3) How many bits are contained within each micro-instruction?
Each micro-instruction contains 18 bits (the 16 control bits and the 2 AdrCtl bits).

c) (3) What is the total number of micro-instructions contained in the control memory for the micro-programmed system for our MIPS core instruction subset?
The control memory contains 10 micro-instructions (one for each of the 10 states).

d) (5) Use hex to show each micro-instruction in the sequence of micro-instructions (i.e., the micro-program) needed for the add machine instruction.
These correspond to the FSM states 0, 1, 6 and 7.



State number	Control word bits 17–2	Control word bits 1–0
0	1001010000001000	11
1	0000000000011000	01
2	0000000000010100	10
3	0011000000000000	11
4	0000001000000010	00
5	0010100000000000	00
6	0000000001000100	11
7	0000000000000011	00
8	0100000010100100	00
9	1000000100000000	00

0x25023
0x00061
0x00113
0x0000C

7. Answer the following questions about the finite state machine (FSM) model of our muti-cycle datapath control shown in module 5:

a) (3) How many separate states are there in the FSM?
There are 10 states (0 through 9).

b) (3) How is the opcode within each machine instruction used by the FSM?
The opcode is used in state 1 and in state 2 to determine which state to enter next.

c) (3) What is the maximum number of states used in the FSM for any of the instructions within our MIPS core instruction subset?
The maximum number is 5 used for the lw instruction.

d) (3) Is the FSM an example of a Mealy machine or an example of a Moore machine (as defined in module 5 CU_implementation)?
The outputs are the control bits, which depend only on the current state, so the FSM is a Moore machine.

8. The ALU shown in the multi-cycle datapath diagram in module 5 has two data input ports: the upper A input port and the lower B input port.

a) (5) Name each of the possible A inputs to the ALU?
The 2 possible A inputs are the rs register and the PC register.

b) (5) Name each of the possible B inputs to the ALU?
The 4 possible B inputs are the constant 4, the rt register, the sign-extended immediate and the sign-extended immediate shifted left 2 bits.

9. (5) Our MIPS ALU outputs a single-bit flag called “zero” that is set to 1 to indicate when the ALU result is 0. Suppose that the ALU also output three other flags which, when set, indicate one of the following conditions:

- N to indicate a negative result
- V to indicate an arithmetic overflow
- C to indicate a carry

After the instruction sub $t0,$t1,$t2 subtracts $t2 from $t1 and places the result into $t0, some combination of the flags (Z, N, V, C) indicates that $t1 contained a value less than that in $t2? Write down the logic expression that shows the required combination of Z, N, V and C in simplest form.
Logic expression = N * V’ + N’ * V = N XOR V.
If there is no overflow, the result must be negative, or positive with an overflow. We know, of course, that if $t1 < $t2 then $t1 - $t2 < 0. However a signed overflow is possible which causes the sign of the value in the result register $t0 to be incorrect.

10. You know that every instruction takes one clock cycle on the single-cycle datapath. How many clock cycles are required to fetch and execute each of the following instructions on our MIPS multi-cycle datapath?

a) (2) sw 4 cycles

b) (2) add 4 cycles

c) (2) slt 4 cycles

d) (2) lw 5 cycles

e) (2) beq 3 cycles

This can be determined from the FSM for the multi-cycle datapath.