Chapter 05: Basic Processing Units … Control Unit Design

Lesson 17:

Microprogram sequencing and next address field

Objective

- Learn microprogram sequencing
- Learn how to use of next address field for generating next instruction address

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Encoding of control signals at control memory address

- When the control signals in a microinstruction are stored in control memory with encoding, then the number of storage bits at each address can be reduced as per the design
- A design selects how many bits to encode out of the needed ones and how many groups of the storing units connect the MUXs

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Encoding of control signals at control memory address

- Consider in a processor thirty two ($=2^6$) storing-units
- Suppose each unit is designed independently with no MUX in between
- Then six bits can encode up to 2^6 storing units controlling signals

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Encoding of control signals at control memory address

- The sixteen select functions encode by 4-bits
- The eight data routes encode by 3-bits
- The total number of output word bits at control memory reduces to 6 + 4 + 3 = 13

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Microprogram sequencing

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Microprogram sequencing

- Several operations execute with varying addressing modes
- For example, consider ADD *r*1, *r*2, *r*3; ADD (*r*1), *r*2, *r*3; ADD *x*(*r*1), *r*2, *r*3; and ADD (*r*1)+, *r*2, *r*3
- The first has register-addressing mode, the second has register-indirect addressing, the third has index-offset addressing mode, and the fourth has post auto-increment addressing mode instructions

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Microprogram Sequencing

- Sets of microinstructions (functions) in Section 5.14.5 used in these instructions
- When for every instruction's microprogram, there is a separate set of sequentially stored microinstructions, the number of addresses needed for all the instructions' microprograms will be too large

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Set of micro-instructions for ADD in four addressing modes

- Func1: Fetch instruction opcode and operands
- Func2: IR $\rightarrow$ ID
- Func3: r2 $\rightarrow$ X, r3 $\rightarrow$ Y,
- Func4: Addition of X and Y and result in Z
- Func5: MDR $\leftarrow$ Z
- Func6: r1 $\rightarrow$ X, *x* $\rightarrow$ Y, Temp $\leftarrow$ Z and Branch Func8
- Func7: Temp $\leftarrow$r1, r1 $\rightarrow$ X, const 4 $\rightarrow$ Y, r1 $\leftarrow$ Z (Post Increment r1 by 4 for next word in memory)
- Func8: Store and fetch next instruction
- Function 9:

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Subsets of the microinstructions Func1 to Func4

- All four ADD instruction's addressing modes execute subset of the microinstructions Func1 to Func4
- There is a way in which the sequence of microinstructions can be changed; just as branch to subroutines at a program in the system memory and different microgram can overlap at the control-store memory

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Example

- Assume control store memory has 12-bit addresses (4096) and therefore instruction decoder will have 12-bits to reflect the starting address of a microprogram

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Example

- Let function select field encode microinstruction branch (MBR bits) as 4 bits = 0110 at the function select field

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Example

- Assume that an instruction control-store address on branching just change has six lower bits of addresses
- A microprogram maximum length assumed as $2^6 = 64$

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Branch microinstruction encoding

- MBR 101000, MBR 101111, MBR 111000 can specify (MBR) to three MBR functions, Func6, Func7, and Func8, respectively
- The last six bits are ORed with lower 6 bits in ID to get the MBR destination address
- If ID bits are 1100 1000 0000, then MBR destination addresses after ORing will be 1100 101 01000, 1100 101 01111, and 1100 101 11000, respectively

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Set of micro-instructions Microinstructions

- Func1: Fetch instruction opcode and operands
- Func2: IR $\rightarrow$ ID
- Fnct3: r3 $\rightarrow$ X, r3 $\rightarrow$ Y, Z $\leftarrow$ X. XOR. Y, Z $\leftarrow$ X
- Fnct4: r3 $\rightarrow$ X, const 1 $\rightarrow$ Y, Z $\leftarrow$ X. ADD. Y, r3 $\leftarrow$ Z (r3 now has two's complement of earlier r3)
- Func5: MDR $\leftarrow$ Z
- Func6: r2 $\rightarrow$ X, r3 $\rightarrow$ Y, Z $\leftarrow$ X. ADD. Y, r1 $\leftarrow$ Z, Branch Func1

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

ADD r1, r2, r3,

- Func1— Fetch instruction opcode and operands
- Func2 — Branch to Func6
- Func6

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

SUB r1, r2, r3 Microinstructions

- Func1─ Fetch instruction opcode and operands
- Function 2─ Branch to Func3
- Func3─ r3 → X, r3 → Y, Z ← X. XOR. Y, Z ← X
- Func4─ r3 → X, const 1 → Y, Z ← X. ADD. Y, r3 ← Z (r3 now has two's complement of earlier r3)
- Function 5─ Branch to Func6
- Func6─ r2 → X, r3 → Y, Z ← X. ADD. Y, r1 ← Z, Branch Func1

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Microinstructions with next address field

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Branch instruction

- Wastes one cycle
- Many microinstructions having common functions to implement
- Therefore, to reduce the number of cycles consumed by branch instruction, another alternative strategy is to place the next address field at each microinstruction

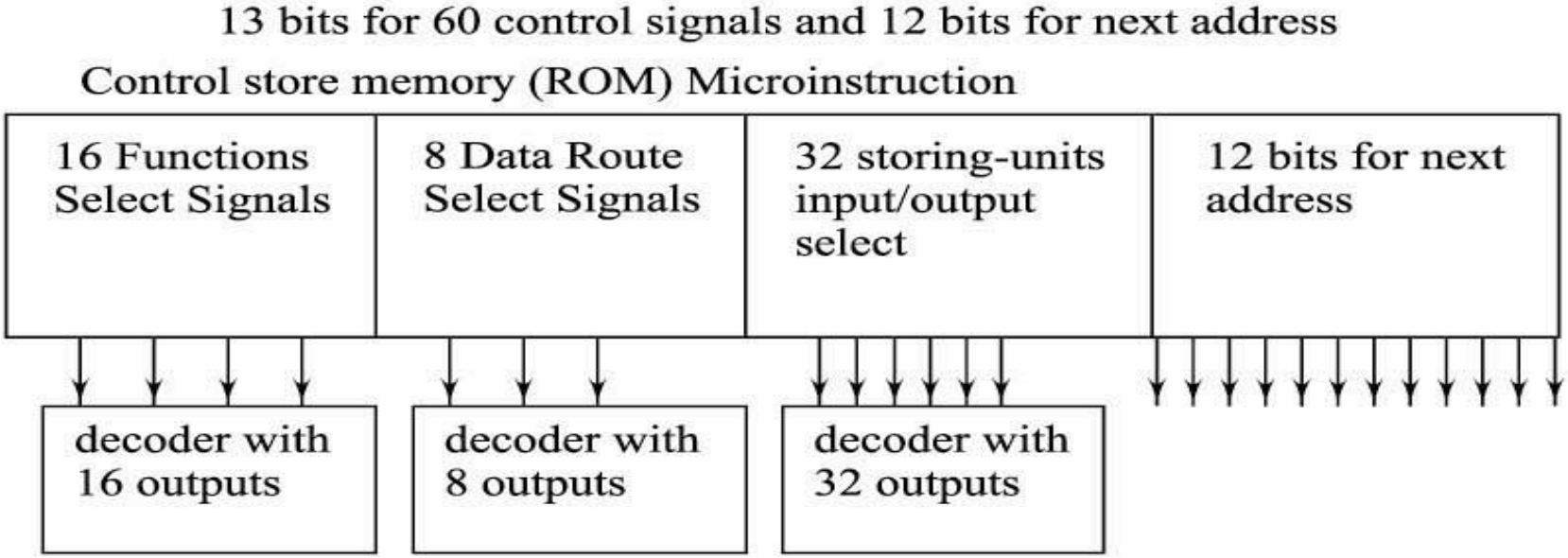
Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Fields of microinstructions with the next address field

- The first time when the microprogram execution starts, the next address is as per ID output, which reflects the IR, condition flags, and external inputs during an instruction
- Then next address field directs it to next microinstruction

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Four fields at a control memory address in a vertical organisation



Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

Summary

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

We Learnt

- Use of horizontal and vertical organization require large number of control bits
- Uses of MUXs in Vertical organisation of Microinstructions
- Microprogram sequencing and overlapping of microinstructions
- Use of next address field for generating next instruction address

Copyright © The McGraw-Hill Companies Inc. Indian Special Edition 2009

End of Lesson 17 on

Microprogram sequencing and next address field