Lecture 19

(UNIT-II)

Reducing length of Micro instructions

- A straightforward way to structure Microinstruction is to assign one bit position to each control signal as shown in Fig. 7.15
- Drawback: assigning individual bits to each control signal results in long microinstructions
- only a few bits are set to 1 (to be used for active gating) in any given microinstruction, which means the available bit space is poorly used.
- Most signals are not needed simultaneously, and many signals are mutually exclusive.
  - Read and Write signals to the memory cannot be active simultaneously.
  - only one function of the ALU can be activated at a time
- The source for a data transfer must be unique

Grouping control signals

- This suggests that signals can be grouped so that all mutually exclusive signals are placed in the same group.
- Thus, at most one *micro-operation* per group is specified in any microinstruction.
- Then it is possible to use a binary coding scheme to represent the signals within a group. For example, four bits suffice to represent the 16 available functions in the ALU.
- It requires little more hardware because decoding circuits must be used to decode the bit patterns of each field into individual control signals.
- Advantage is that the size of the control store is reduced.

Field Encoded Micro-instructions

Microinstruction

F1	F2	F3	F4	F5

F1 (4 bits)	F2 (3 bits)	F3 (3 bits)	F4 (4 bits)	F5 (2 bits)
0000: No transfer	000: No transfer	000: No transfer	0000: Add	00: No action
0001: PC_{out}	001: PC_{in}	001: MAR_{in}	0001: Sub	01: Read
0010: MDR_{out}	010: IR_{in}	010: MDR_{in}	⋮	10: Write
0011: Z_{out}	011: Z_{in}	011: $TEMP_{in}$		
0100: $R0_{out}$	100: $R0_{in}$	100: Y_{in}	1111: XOR	
0101: $R1_{out}$	101: $R1_{in}$		16 ALU functions	
0110: $R2_{out}$	110: $R2_{in}$			
0111: $R3_{out}$	111: $R3_{in}$			
1010: $TEMP_{out}$				
1011: $Offset_{out}$				

F6	F7	F8	···

F6 (1 bit)	F7 (1 bit)	F8 (1 bit)
0: SelectY	0: No action	0: Continue
1: Select4	1: WMFC	1: End

Figure 7.19 An example of a partial format for field-encoded microinstructions.

Horizontal Organization of microinstructions

- minimally encoded scheme of Fig 7.15, in which many resources can be controlled with a single microinstruction, is called a *horizontal organization*.
- horizontal microinstructions have larger lengths
- Involves enumerating the patterns of required signals in all possible microinstructions.
- The horizontal approach is useful when a higher operating speed is desired and when the machine structure allows parallel use of resources.
- Each meaningful combination of active control signals can then be assigned a distinct code that represents the microinstruction.

Micro - instruction	..	PC_{in}	PC_{out}	MAR_{in}	Read	MDR_{out}	IR_{in}	Y_{in}	Select	Add	Z_{in}	Z_{out}	$R1_{out}$	$R1_{in}$	$R3_{out}$	WMFC	End	..
1		0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	
2		1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	
3		0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	
4		0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	
5		0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	
6		0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	
7		0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	

Figure 7.15 An example of microinstructions for Figure 7.6.

Vertical Organization of microinstructions

- Highly encoded schemes that use compact codes to specify only a small number of control functions in each microinstruction are referred to as a *vertical organization*.
- vertical microinstructions have smaller lengths
- The vertical approach results in considerably slower operating speeds because more microinstructions are needed to perform the desired control functions.
- The significant factor is that less hardware is needed to handle the execution of microinstructions.

Hardwired vs. Micro-programmed Computers

SN	Hardwired Control	Micro-programmed Control
1	***Difficult to design**: Composed of complex combinatorial and sequential circuits that generate complete timing that corresponds with execution of each instruction.	***Simpler to design:*** The process of specifying the architecture and instruction set is now one of software (micro-programming) as opposed to hardware design.
2	***Difficult to modify***: Once the control unit of a hard-wired computer is designed and built, it is virtually impossible to alter its architecture and instruction set. To do this it requires a complete redesign of the controller circuit hardware.	***Easier to modify*:** Microprogramming offers flexibility for design and architectural changes. The control memory (ROM) can be reprogrammed or replaced. We can change the computer's instruction set simply by altering the micro-program stored in its control memory.
3	***Faster than Micro-programmed Control***: If speed is a consideration, hard-wiring may be required since it is faster to have the hardware issue the required control signals than to have a "program" do it. Hardwired control is fast because the cycle time depends on the combinational logic delay of the control unit, which is much less than memory access time.	***Slower than hardwired control:*** Microprogramming is slow because the control memory is accessed in every cycle, and the memory access is slow.
4.	Used in RISC computers	Used in CISC computers