## Computer Architecture

Prof. Dr. Nizamettin AYDIN

naydin@yildiz.edu.tr
nizamettinaydin@gmail.com

http://www.yildiz.edu.tr/~naydin

1

## Computer Architecture

# Micro-Programmed Control

2

## Outline

- Micro-Programmed Control
  - Control Unit Implementation
    - Hardwired Implementation
    - Micro-programmed Control
  - Control Unit Organization
  - Micro-instruction Types
  - Organization of Control Memory
  - Sequencing Techniques
  - Address Generation
  - Microinstruction Execution
  - Control Unit Organization
  - Microinstruction Encoding

3

## Control Unit Implementation

- Hardwired implementation
  - Combinatorial circuit
- Microprogrammed implementation

4

## Hardwired Implementation (1)

- Control unit inputs
- Flags and control bus
  - Each bit means something
- Instruction register
  - Op-code causes different control signals for each different instruction
  - Unique logic for each op-code
  - Decoder takes encoded input and produces single output
  - *n* binary inputs and $2^n$ outputs

5

## Hardwired Implementation (2)

- Clock
  - Repetitive sequence of pulses
  - Useful for measuring duration of micro-ops
  - Must be long enough to allow signal propagation
  - Different control signals at different times within instruction cycle
  - Need a counter with different control signals for t1, t2 etc.

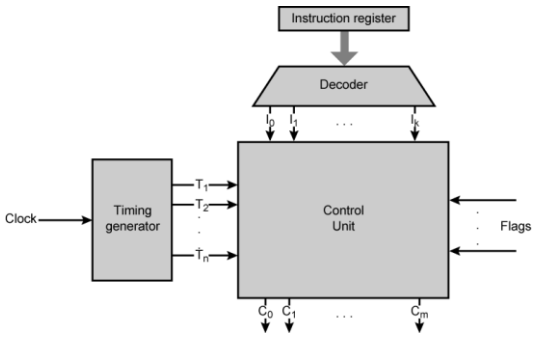
6

## A Decoder with four inputs and 16 outputs

| I1 | I2 | I3 | I4 | O1 | O2 | O3 | O4 | O5 | O6 | O7 | O8 | O9 | O10 | O11 | O12 | O13 | O14 | O15 | O16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



## Control Unit with Decoded Inputs






## Control signal example

| Micro-operations | Timing | Active Control Signals |
|---|---|---|
| Fetch: | $t_1$: MAR ← (PC) | $C_2$ |
| | $t_2$: MBR ← Memory<br>PC ← (PC) + 1 | $C_5$, $C_R$ |
| | $t_3$: IR ← (MBR) | $C_4$ |
| Indirect: | $t_1$: MAR ← (IR(Address)) | $C_8$ |
| | $t_2$: MBR ← Memory | $C_5$, $C_R$ |
| | $t_3$: IR(Address) ← (MBR(Address)) | $C_4$ |
| Interrupt: | $t_1$: MBR ← (PC) | $C_1$ |
| | $t_2$: MAR ← Save-address<br>PC ← Routine-address | |
| | $t_3$: Memory ← (MBR) | $C_{12}$, $C_W$ |

$C_R$ = Read control signal to system bus.
$C_W$ = Write control signal to system bus.

- Boolean expression for $C_5$
  - $C_5 = P' \cdot Q' \cdot T_2 + P' \cdot Q \cdot T_2$
- C5 is also needed during execute cycle
  - Let us assume there are only three instructions
    - LDA, ADD, AND
- Boolean expression for C5 becomes

  $C_5 = P' \cdot Q' \cdot T_2 + P' \cdot Q \cdot T_2 + P \cdot Q' \cdot (LDA + ADD + AND)$

- Consider a single signal, $C_5$.
  - causes data to be read from the external data bus into the MBR
- Used twice in the table
- Let us define two new control signals, P and Q:

  PQ = 00 Fetch cycle
  PQ = 01 Indirect cycle
  PQ = 10 Execute cycle
  PQ = 11 Interrupt cycle



## Problems With Hard Wired Designs

- Complex sequencing & micro-operation logic
- Difficult to design and test
- Inflexible design
- Difficult to add new instructions



## Micro-programmed Control

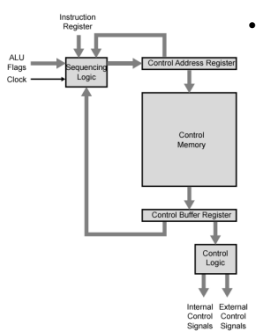
| Micro-operations | Timing | Active Control Signals |
|---|---|---|
| Fetch: | $t_1$: MAR ← (PC) | $C_2$ |
| | $t_2$: MBR ← Memory<br>PC ← (PC) + 1 | $C_5$, $C_R$ |
| | $t_3$: IR ← (MBR) | $C_4$ |
| Indirect: | $t_1$: MAR ← (IR(Address)) | $C_8$ |
| | $t_2$: MBR ← Memory | $C_5$, $C_R$ |
| | $t_3$: IR(Address) ← (MBR(Address)) | $C_4$ |
| Interrupt: | $t_1$: MBR ← (PC) | $C_1$ |
| | $t_2$: MAR ← Save-address<br>PC ← Routine-address | |
| | $t_3$: Memory ← (MBR) | $C_{12}$, $C_W$ |

$C_R$ = Read control signal to system bus.
$C_W$ = Write control signal to system bus.

- Consider the following table
  - Notation looks like a programming language
- In fact it is a language
  - Called microprogramming language
- Each line describes a set of micro-operations
  - known as a microinstruction
- A sequence of microinstructions
  - Known as microprogram
  - or firmware



## Control Unit Organization




- All the control unit (CU) does is to generate a set of control signals
  - Each control signal is reprented by a bit
    - which is either on or off
  - CU have a control word for each micro-operation
  - CU have a sequence of control words for each machine code instruction
  - CU adds an address to specify the next micro-instruction, depending on conditions

## Control Unit Organization

- Today's large microprocessor
  - Many instructions and associated register-level hardware
  - Many control points to be manipulated
- This results in control memory that
  - Contains a large number of words
    - corresponding to the number of instructions to be executed
  - Has a wide word width
    - Due to the large number of control points to be manipulated

13

## Micro-program Word Length

- Based on 3 factors
  - Maximum number of simultaneous micro-operations supported
  - The way control information is represented or encoded
  - The way in which the next micro-instruction address is specified

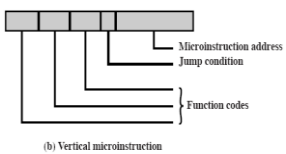
14

## Micro-instruction Types

- Vertical micro-programming
  - Each micro-instruction specifies single (or few) micro-operations to be performed
- Horizontal micro-programming
  - Each micro-instruction specifies many different micro-operations to be performed in parallel

15

## Vertical Micro-programming

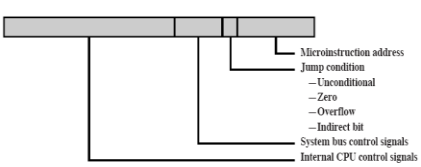


(b) Vertical microinstruction

- Width is narrow
- n control signals encoded into $\log_2 n$ bits
- Limited ability to express parallelism
- Considerable encoding of control information requires external memory word decoder to identify the exact control line being manipulated

16

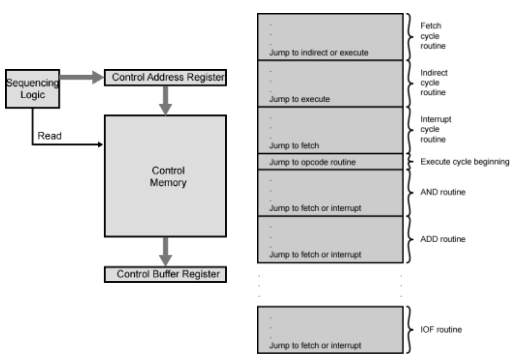
## Horizontal Micro-programming




- Wide memory word.
  - 1 bit for each internal processor control line
  - 1 bit for each system bus control line
  - Condition field
  - Addres field
- High degree of parallel operations possible
- Little encoding of control information

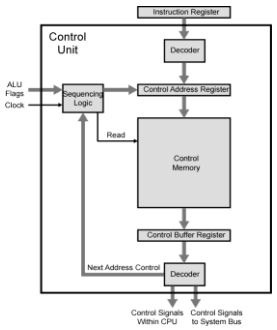
17

## Organization of Control Memory




18

## Control Unit Function




- Sequence logic unit issues read command
- Word specified in control address register is read into control buffer register
- Control buffer register contents generates control signals and next address information
- Sequence logic loads new address into control buffer register based on next address information from control buffer register and ALU flags

19

## Next Address Decision

- Depending on ALU flags and control buffer register
  - Get next instruction
    - Add 1 to control address register
  - Jump to new routine based on jump microinstruction
    - Load address field of control buffer register into control address register
  - Jump to machine instruction routine
    - Load control address register based on opcode in IR

20

## Pros and Cons

- Advantages of Microprogramming
  - Simplifies design of control unit
    - Cheaper
    - Less error-prone
- Disadvantages of Microprogramming
  - Slower

21

## Tasks Done By Microprogrammed Control Unit

- Microinstruction sequencing
  - Get the next microinstruction from the control memory
- Microinstruction execution
  - Generate the control signals needed to execute the microinstruction
- In designing the control unit, these tasks must be considered together,
  - because both affect the format of the microinstruction
  - and the timing of control unit

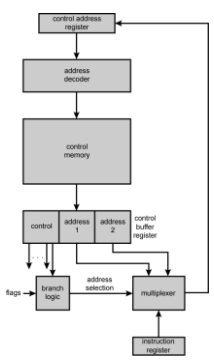
22

## Sequencing Techniques

- Generation of the control memory addres for the next instruction is based on...
  - current microinstruction,
  - condition flags, and
  - contents of IR,
- A number of categories exist.
- These categories are based on...
  - format of address information:
    - Two address fields
    - Single address field
    - Variable format

23

## Branch Control Logic: Two Address Fields

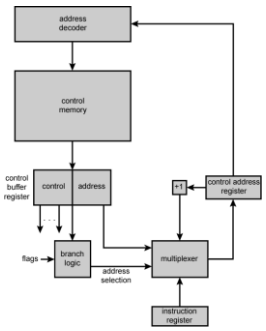



- A multiplexer (MUX) ...
  - serves as a destination for..
    - both address fields, and
    - IR
- Based on an address selection input, MUX transmits either...
  - the op-code or
  - one of the two addresses

  to CAR
- CAR is subsequently decoded to produce the next microinstruction address
- The address selection signals are provided by a branch logic module,
  - whose input consists of ...
    - CU flags, and
    - bits from the control portion of the microinstruction

24

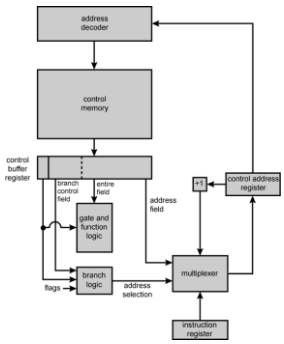
## Branch Control Logic: Single Address Field




- Options for next address:
  - Address field
  - IR code
  - Next sequential address
- Determined by address selection signals
- This approach reduces the number of address fields to one
- Thus, there is some inefficiency in the microinstruction coding scheme

25

## Branch Control Logic: Variable Format




- Two different microinstruction formats are provided:
  - One bit designates which format is being used
- In one format,...
  - remaining bits are used to activate control signals
- In the other format,...
  - some bits drive the branch logic module, and
  - the remaining bits provide the address.
- Next address is either...
  - the next sequential address or
  - an address derived from the IR

26

## Address Generation

- Looked at the sequencing problem from the point of format considerations and general logic requirement
- Another view point is to consider the various ways in which the next address is computed
- Can be divided into two main techniques
  - Explicit techniques
    - Address is explicitly available in the microinstruction
  - Implicit techniques
    - Require additional logic to generate address

27

## Address Generation

- Explicit techniques have been discussed:
  - With a two-field approach, two alternative addresses are available with each microinstruction.
  - Using either a single address field or a variable format, various branch instructions can be implemented.
- A conditional branch instruction depends on the...
  - ALU flags
  - part of the opcode or address mode fields of the machine instruction
  - parts of a selected register, such as the sign bit
  - status bits within the CU

28

## Address Generation

- Three main implicit techniques:
- Mapping
  - Opcode portion of a machine instruction must be mapped into a microinstruction address
  - This occurs only once per instruction cycle
- Adding
  - Two portions of an address are combined to form the complete address
- Residual control
  - Involves the use of a microinstruction address that has previously been saved in temporary storage within the CU.

29

## Address Generation

| Explicit | Implicit |
|---|---|
| Two-field | Mapping |
| Unconditional Branch | Addition |
| Conditional branch | Residual control |

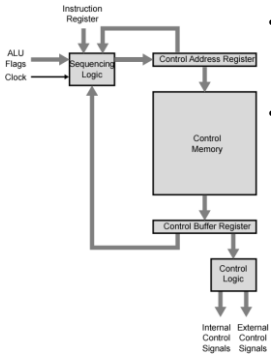
30

## Microinstruction Execution

- The microinstruction cycle is the basic event on a microprogrammed processor
- Each cycle is made up of two events
  - Fetch
    - Determined by generation of microinstruction address
  - Execute
    - Effect is to generate control signals
    - Some control points internal to processor
    - Rest go to external control bus or other interface

31

## Control Unit Organization




- Microinstructions can be classified in a variety of ways
- Distinctions that are commonly made in the literature include...
  - Vertical/horizontal
  - Packed/unpacked
  - Hard/soft microprogramming
  - Direct/indirect encoding

32

## How to Encode

- Consider that there are K different internal and external control signals
- Not all of the possible combinations will be used.
- For example:
  - Two sources cannot be gated to same destination
  - Register cannot be source and destination
  - Only one pattern presented to ALU at a time
  - Only one pattern presented to external control bus at a time
- Require $Q < 2K$ which can be encoded with $\log_2 Q < K$ bits
- In practice this form of encoding is not used for the following reasons:
  - As difficult to program as pure decoded scheme
  - Requires complex slow control logic module
- Instead, some compromises are made as follows:
  - More bits than necessary are used to encode the possible combinations
  - Some combinations that are physically allowable are not possible to encode

33

## Specific Encoding Techniques

- Microinstruction organized as set of fields
- Each field contains code
- Activates one or more control signals
- Organize format into independent fields
  - Field depicts set of actions (pattern of control signals)
  - Actions from different fields can occur simultaneously
- Alternative actions that can be specified by a field are mutually exclusive
  - Only one action specified for field could occur at a time

34

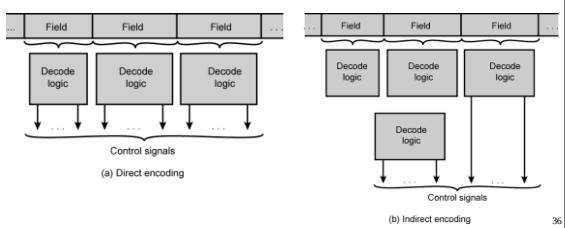
## Organizing encoded micro instruction in to fields

- Two approaches ca be taken to organizing encoded micro instruction in to fields:
  - Functional encoding
    - Identifies functions within the machine and designates fields by function type.
    - For example, if various sources can be used for transferring data ta the accumulator, one field can be designated for this purpose, with each code specifying a different source
  - Resource encoding
    - Views the machine as consisting of a set of independent resources and devotes one field to each (e.g., I/O, memory, ALU)

35

## Microinstruction Encoding
## Direct Encoding

- Another aspect of encoding is whether it is direct or indirect




36