Computer Organization & Architecture

Chapter 5 – Hardwired Control & Microporammed Control

Zhang Yang 张杨

cszyang@scut.edu.cn

Autumn 2024

Content of this lecture

- 5.5 Control Signals
- 5.6 Hardwired Control
- 5.7.2 Microprogrammed Control

Control Signals

- Inter-stage Registers: RA, RB, RZ, RY, RM, and PC-Temp
  - Since data are transferred from one stage to the next in every clock cycle, inter-stage registers are always enabled.
- PC, IR, and the Register File
  - Must be enabled when new data are loaded in.
- MuxB, MuxY
  - To simplify the required control circuit, the same selection can be maintained in all execution steps.
- MuxMA
  - Must change its selection in different execution steps.
  - In Step1:Select PC
  - In Step 4 of Load and Store: Select RZ

Control Signals For the Datapath

- Figure 5.18

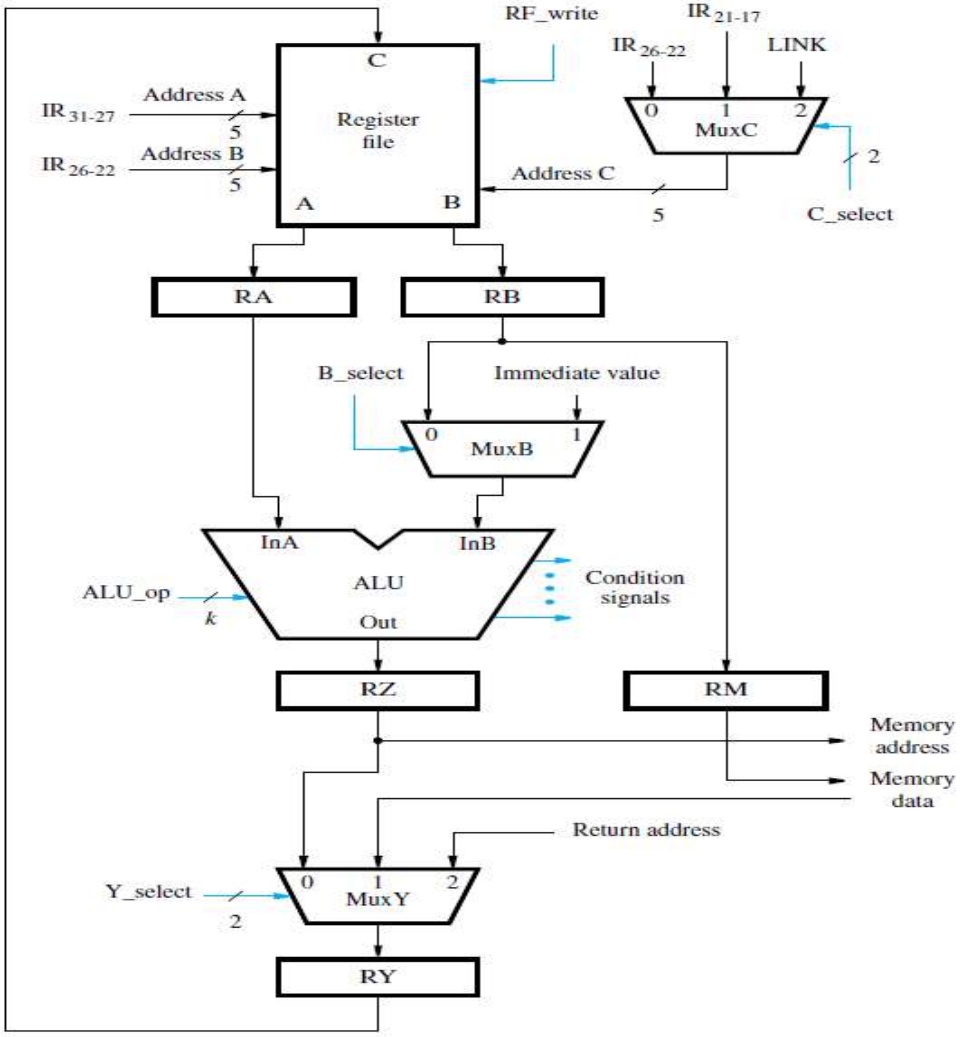


Figure 5.18 Control signals for the datapath.

Recall Instruction Encoding

- Figure 5.12

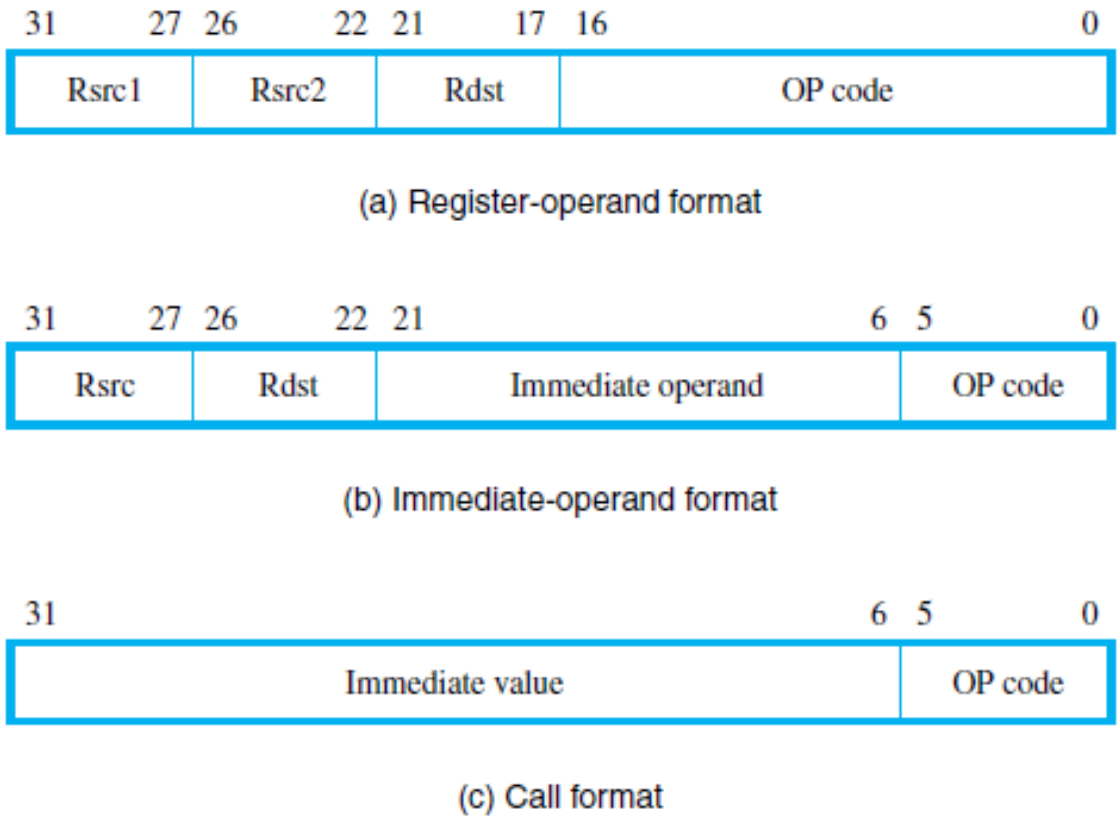
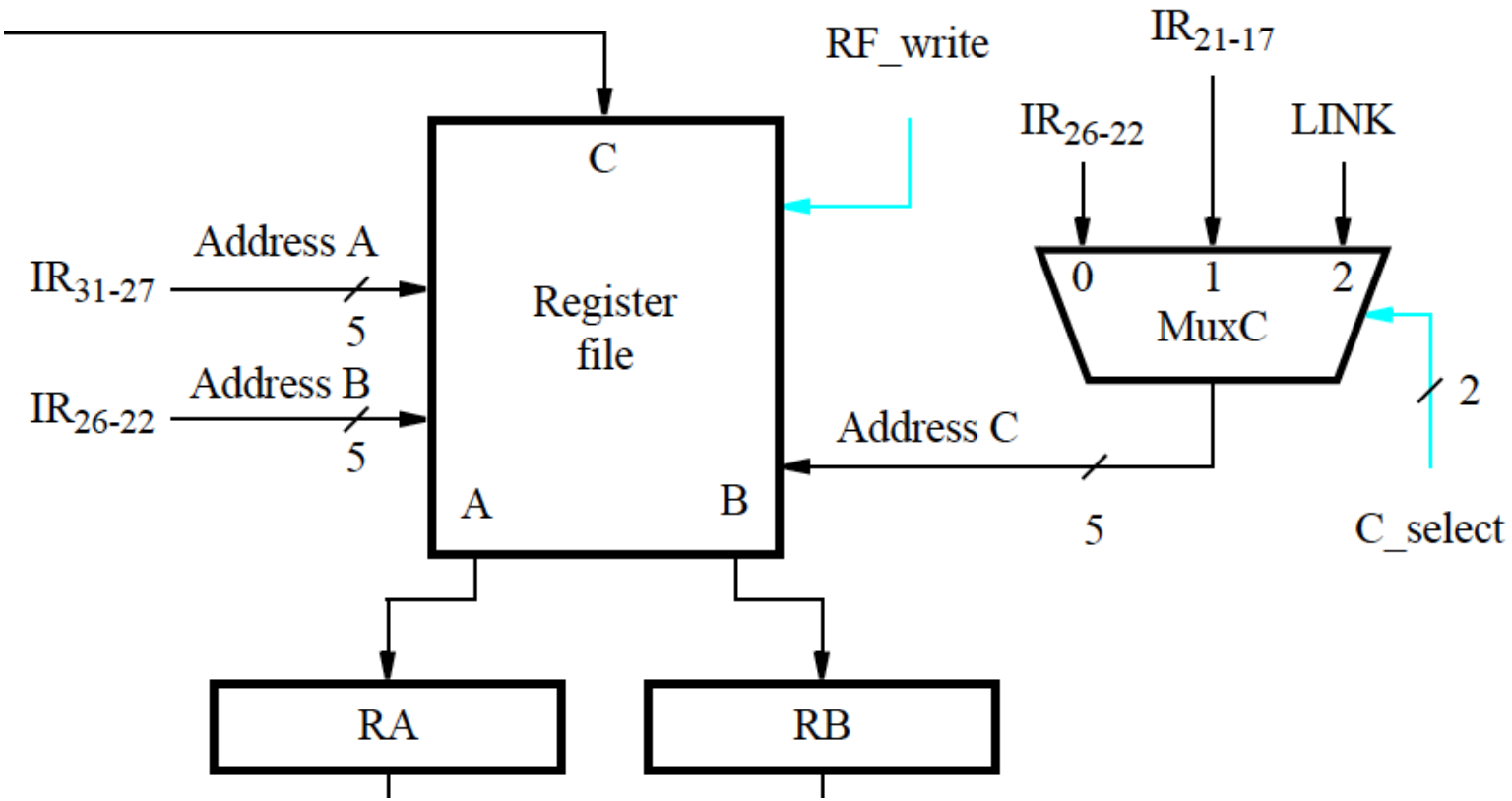


Figure 5.12 Instruction encoding.

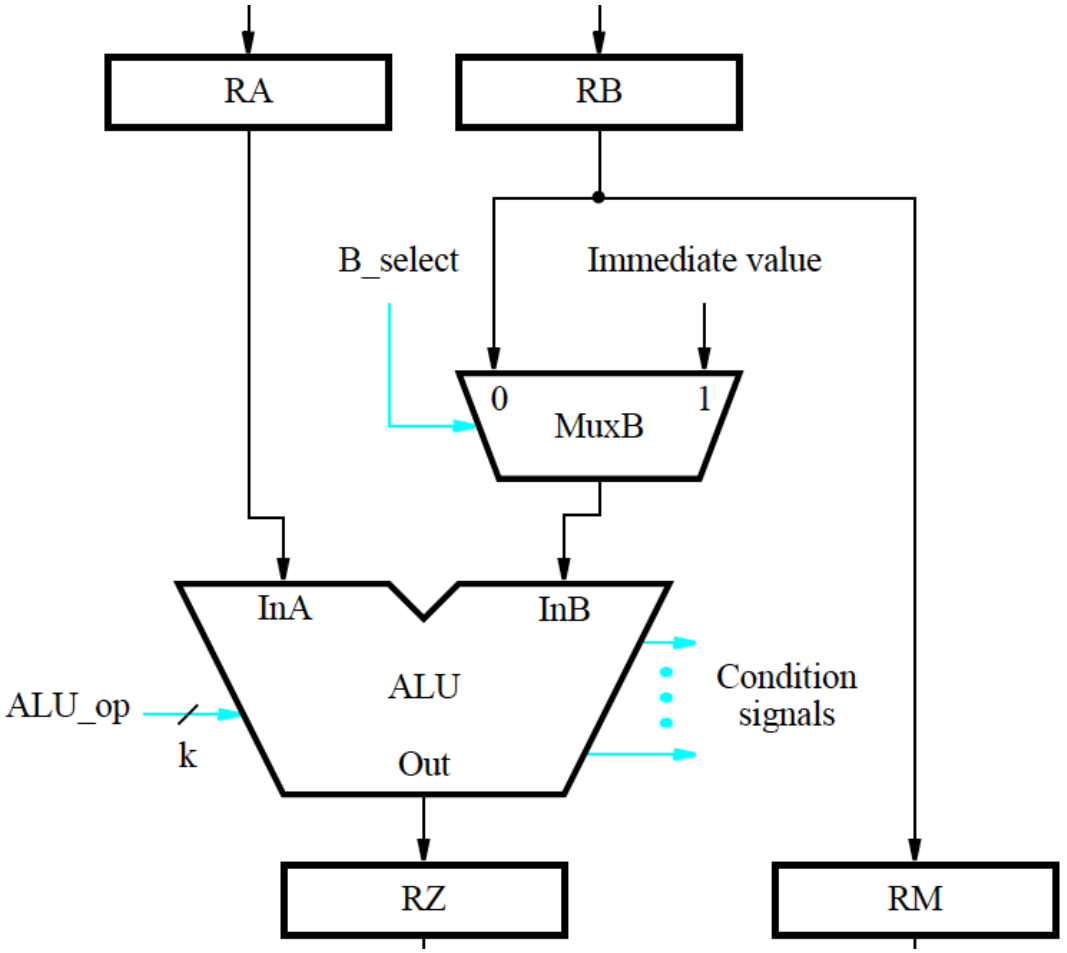
Register File Control Signals

- Part of Figure 5.18
  - RF_write
  - C_select



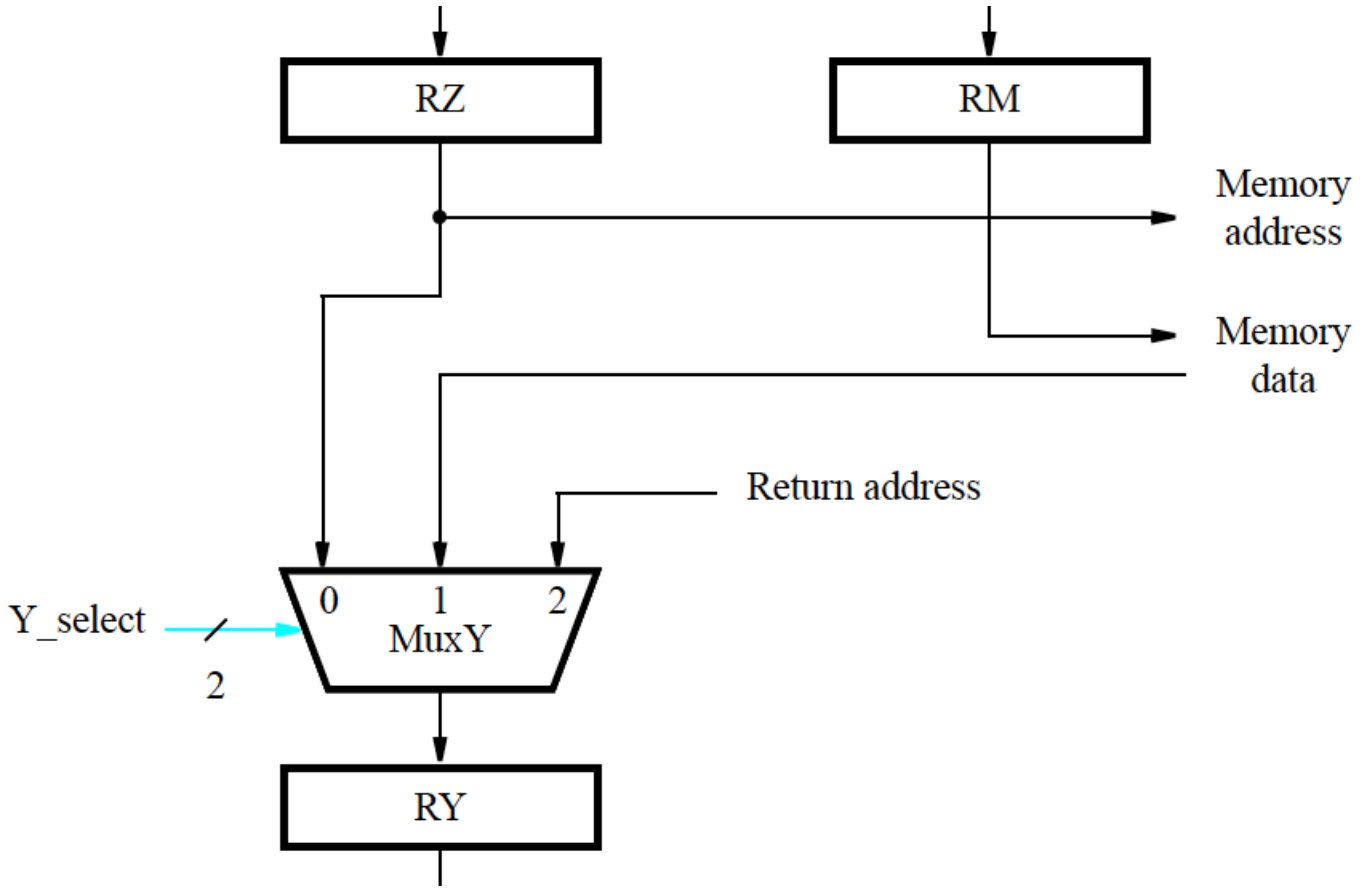
ALU Control Signals

- Part of Figure 5.18 (ctd.)
  - B_selcet
  - ALU_op
  - Condition signals



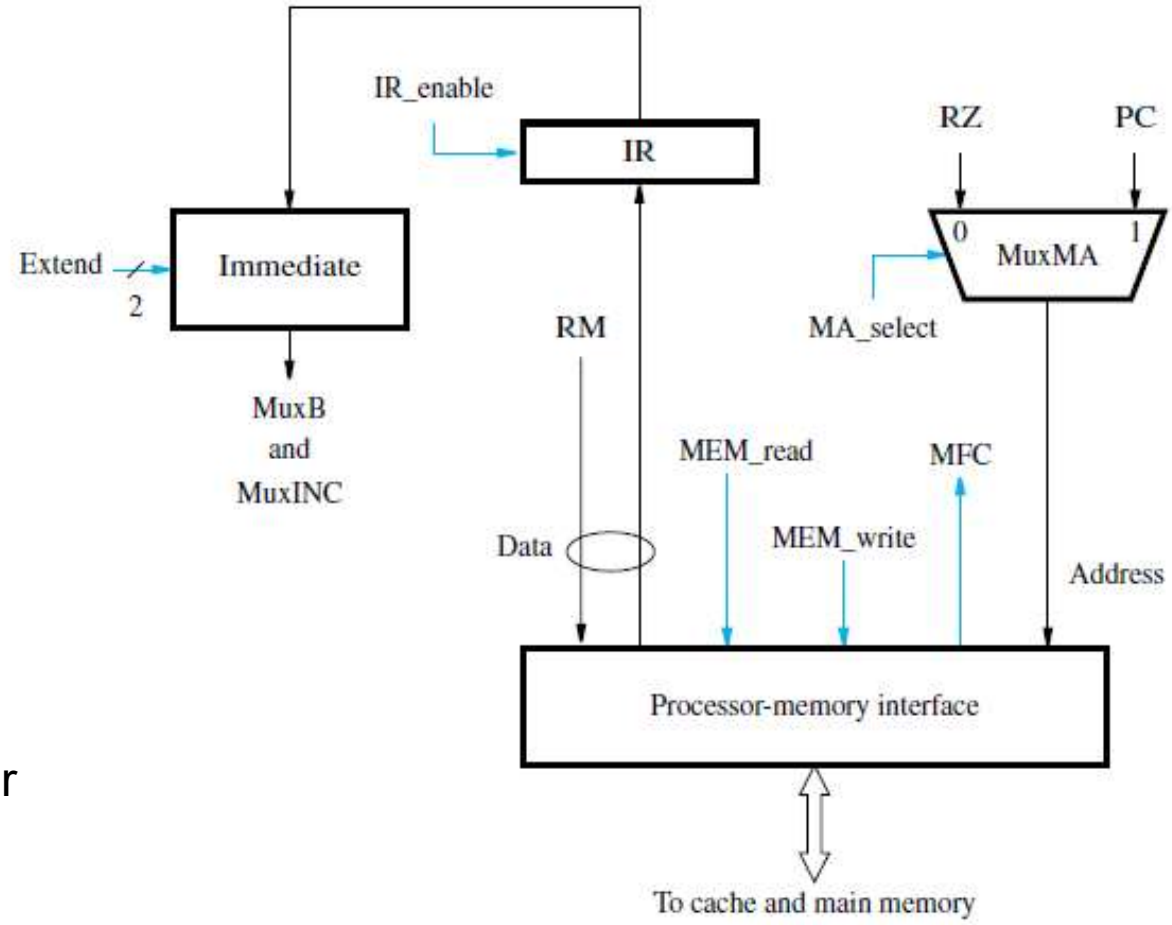
Result Selection Control Signals

- Part of Figure 5.18 (ctd.)
  - Y_select



Processor-Memory Interface Signals

- Figure 5.19
  - MA_select
  - MEM_read
  - MEM_write
  - MFC



MEM_read and MEM_write are used to initiate a memory Read or a memory Write operation.

Figure 5.19 Processor-memory interface and IR control signals.

IR Control Signals (1)

- Figure 5.19
  - IR_enable

During a fetch step, IR_enable must be activated only after the MFC signal is asserted.

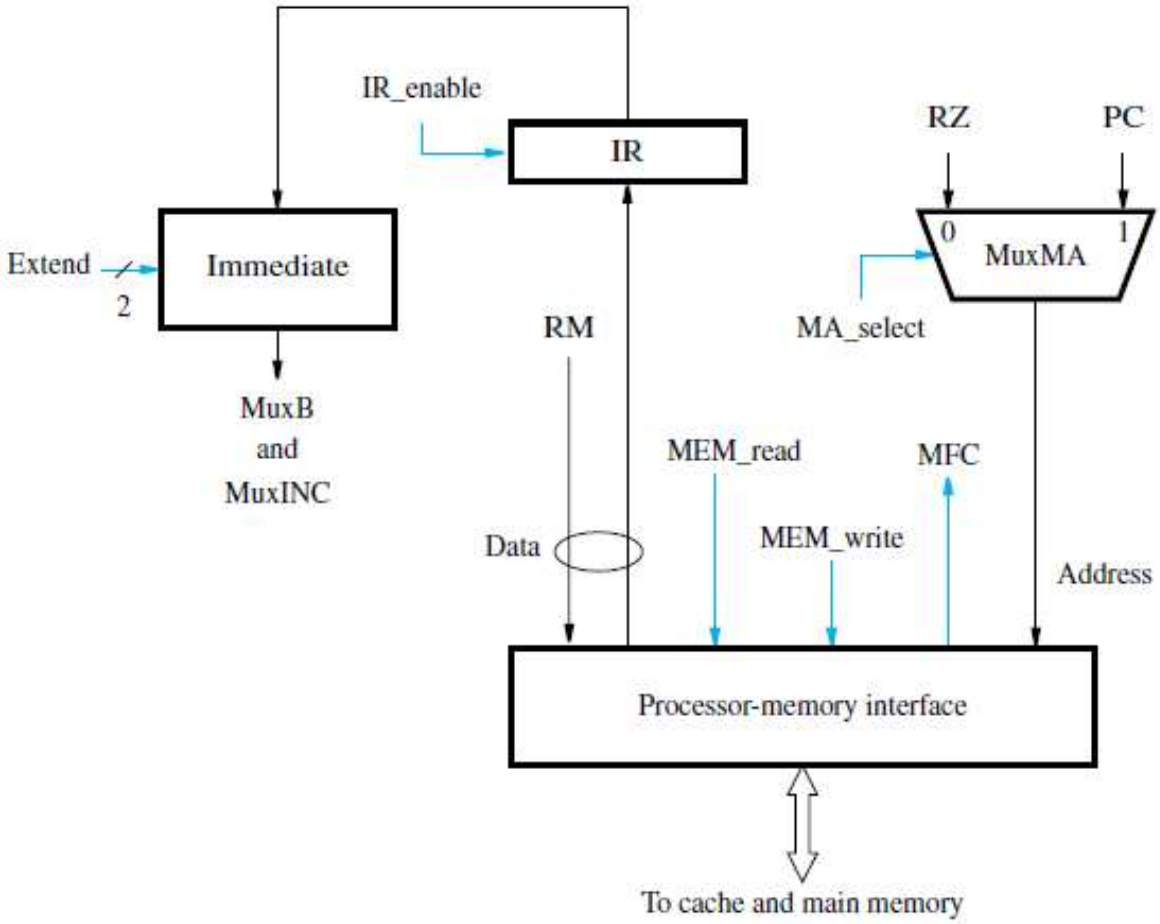


Figure 5.19 Processor-memory interface and IR control signals.

IR Control Signals (2)

- Figure 5.19 (ctd.)
  - Extend

immediate value : be extended to a 32-bit value. Assume that the immediate value is used in three different ways:
(*a*) A 16-bit value is sign-extended for use in arithmetic operations.
(*b*) A 16-bit value is padded with zeros to the left for use in logic operations.
(*c*)A 26-bit value is padded with 2 zeros to the right and the 4 high-order bits of the PC are appended to the left for use in subroutine-call instructions.

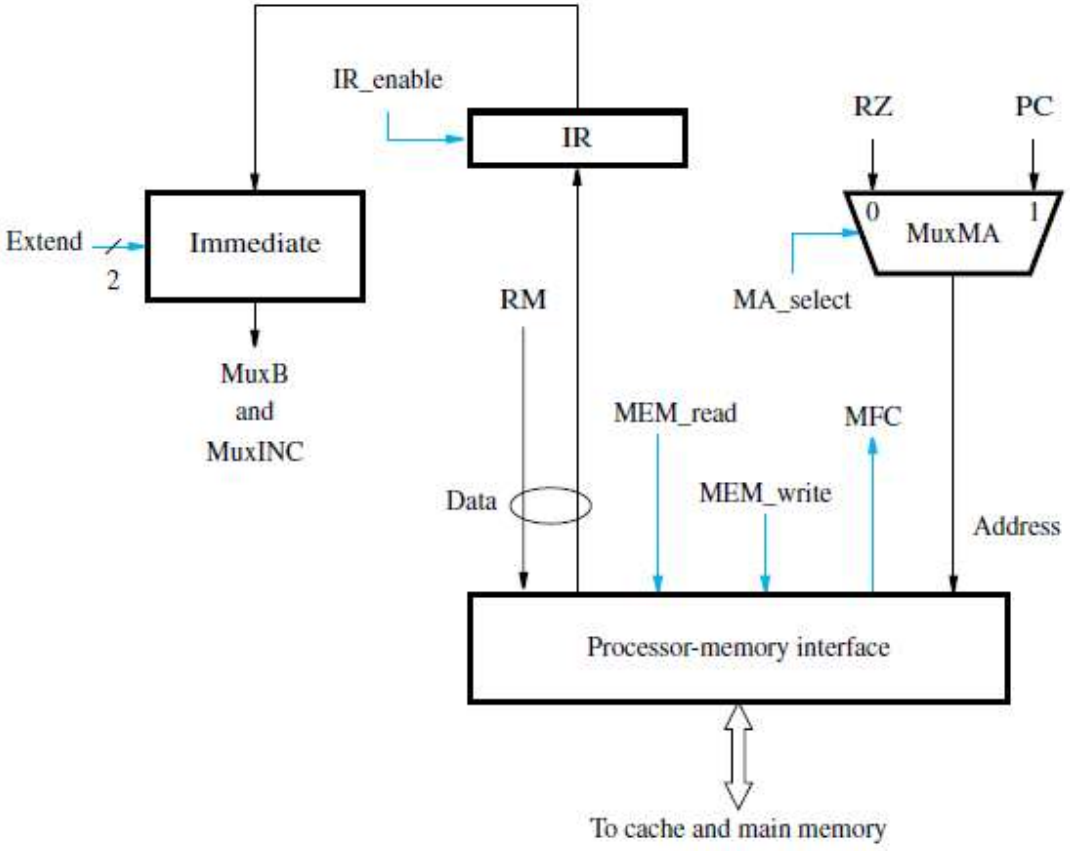


Figure 5.19 Processor-memory interface and IR control signals.

Instruction Address Generator Control Signals

- Figure 5.20
  - PC_select
  - PC_enable
  - INC_select

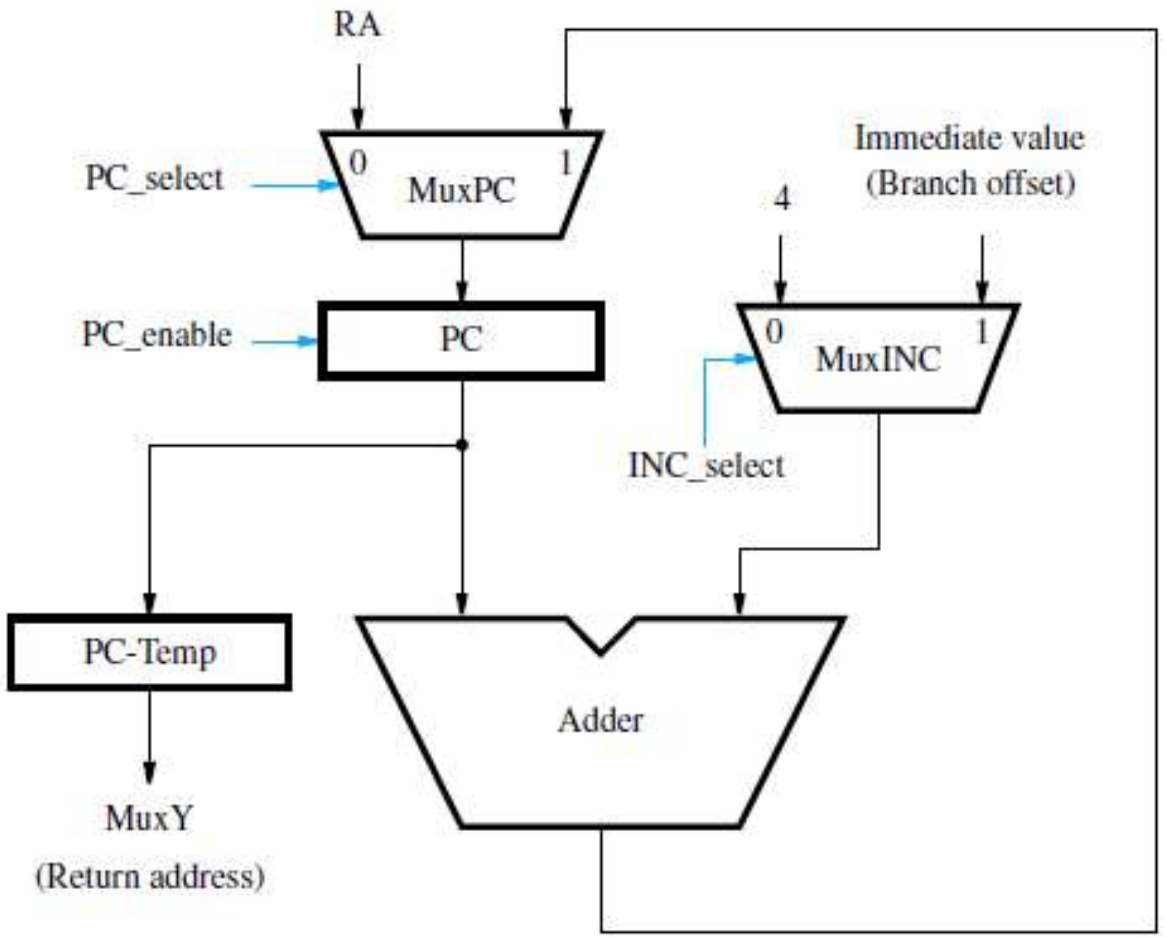


Figure 5.20 Control signals for the instruction address generator.

Summary

- 知识点：Control Signals
- 掌握程度
  - 给出数据通路图，理解各个控制信号。

Content of this lecture

- 5.5 Control Signals
- 5.6 Hardwired Control
- 5.7.2 Microprogrammed Control

Control Signal Generation

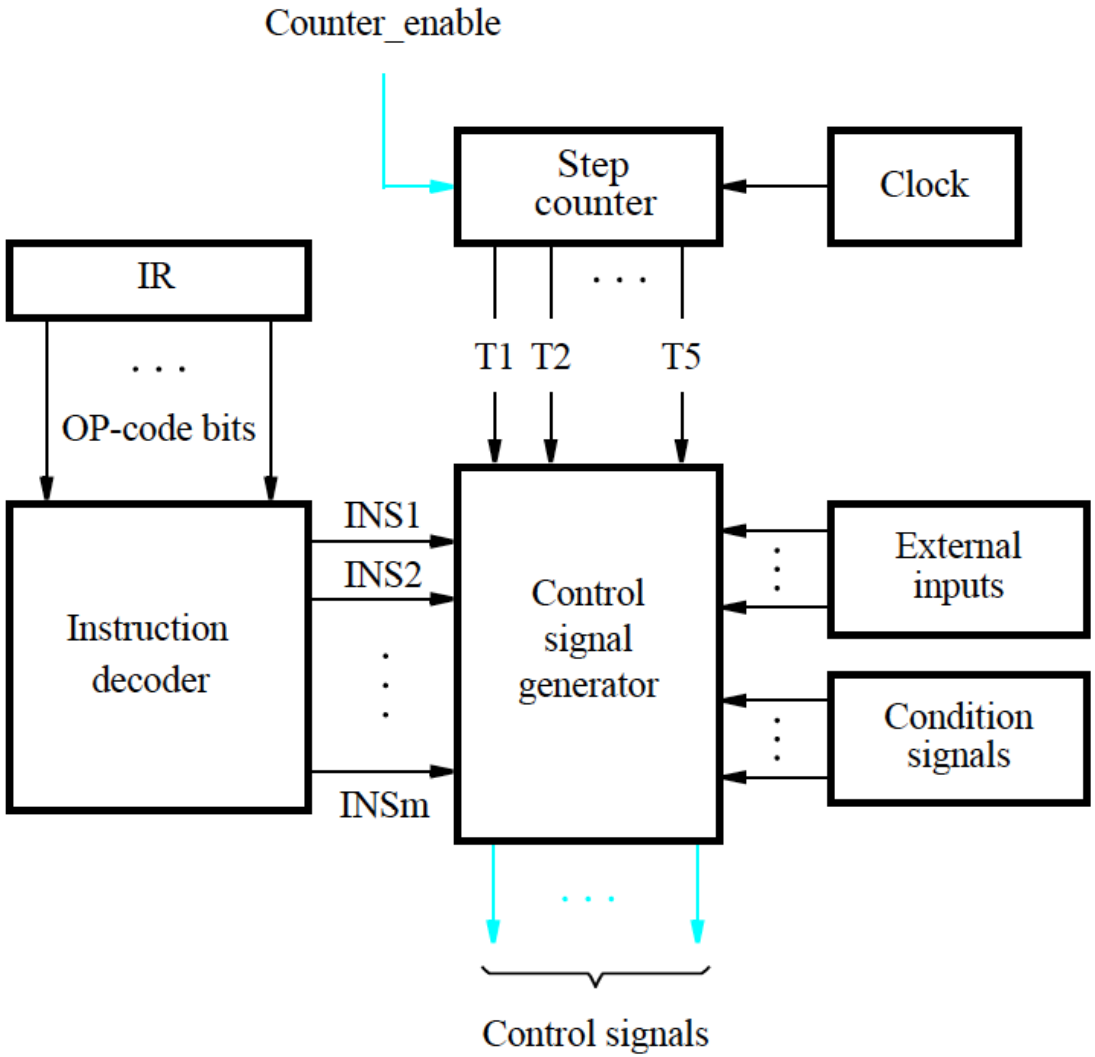
- Circuitry must be implemented to generate control signals so actions take place in correct sequence and at correct time.
- There are two basic approaches
  - Hardwired Control
    - The control unit is essentially a combinational circuit. Its input logic signals are transformed into a set of output logic signals, which are the control signals.
  - Microprogrammed Control
    - The control signals are generated by a microprogram.

Hardwired Control (1)

- Control Signal Generation
  - Contents of the step counter.
  - Contents of the instruction register.
  - The result of a computation or a comparison operation.
  - External input signals, such as interrupt requests.

Hardwired Control (2)

- Control Signal Generation (ctd.)



Hardwired Control Unit Design (1)

- Design Prerequisites
  - Define Control Objects (Output: Control Signals)
  - Identify Input Conditions
  - Define Timing System
- Design Steps
  - Step 1: Specify Instruction Set & Decompose Execution Flows
  - Step 2: Create "Beat-Microoperation" Mapping Table

Hardwired Control Unit Design (2)

- Design Steps (ctd.)
  - Step 3: Derive Logical Expressions for Control Signals
    - Using AND for signals that must be active simultaneously (e.g., opcode + beat);
    - Using OR for signals active in multiple scenarios (e.g., same control signal for different instructions/beats);
    - Simplifying with Boolean algebra or Karnaugh maps to minimize gate count.
  - Step 4: Hardware Implementation
    - Build Combinational Logic Circuits
  - Step 5: Verification & Optimization
    - Resolve Hazards & Ensure Functionality

Instruction Fetch Control Signals (1)

- Recall : Step1 (instruction fetch):

Memory address ← [PC], Read memory, Wait for MFC,
IR ← Memory data, PC ← [PC] + 4

T1=1,
MA_select=1,
MEM_read, WMFC, IR_enable
INC_select=0, PC_select=1, PC_enable=1

Instruction Fetch Control Signals (2)

MA_select=1,

MEM_read, WMFC. IR enable

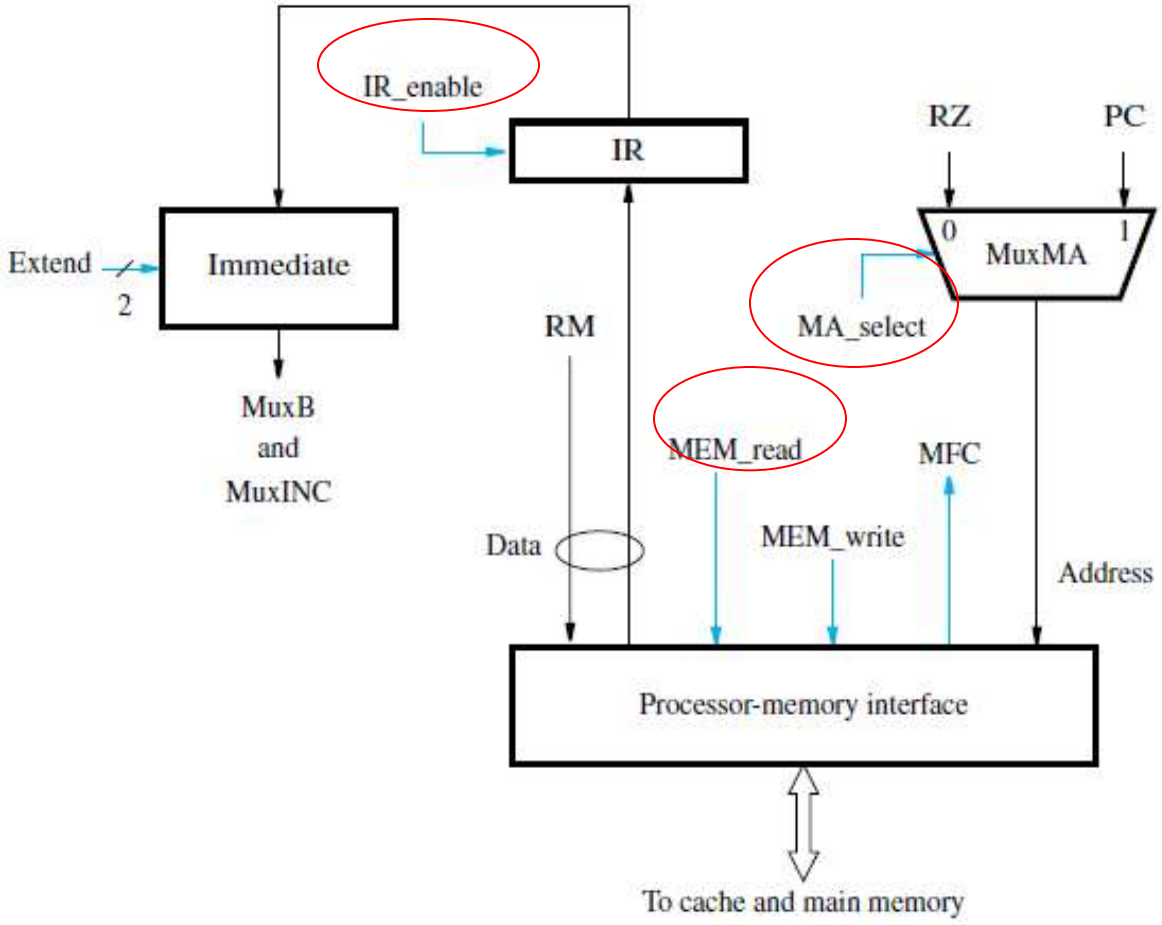


Figure 5.19 Processor-memory interface and IR control signals.

Instruction Fetch Control Signals (3)

INC_select=0, PC_select=1, PC_enable=1

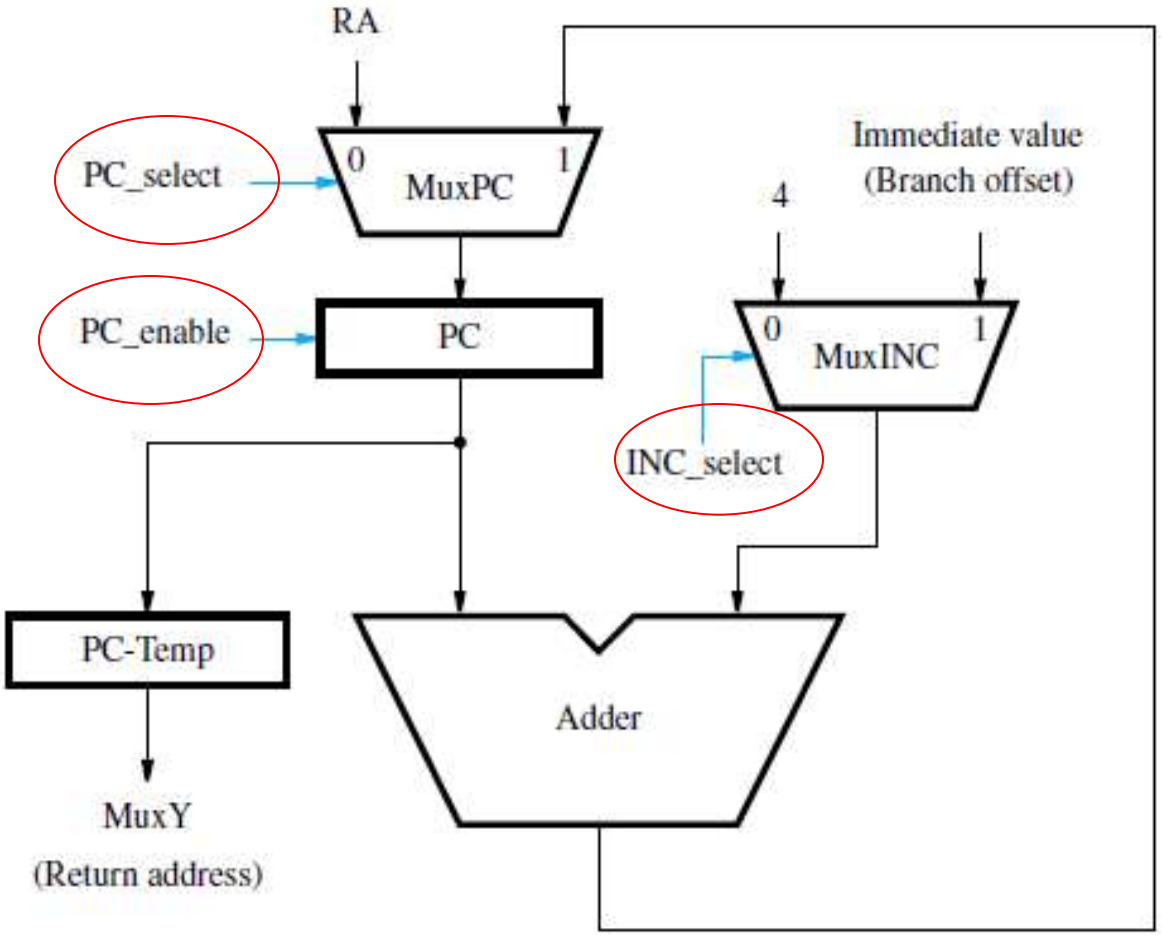


Figure 5.20 Control signals for the instruction address generator.

Data Path Control Signals (1)

- Datapath Control Signals

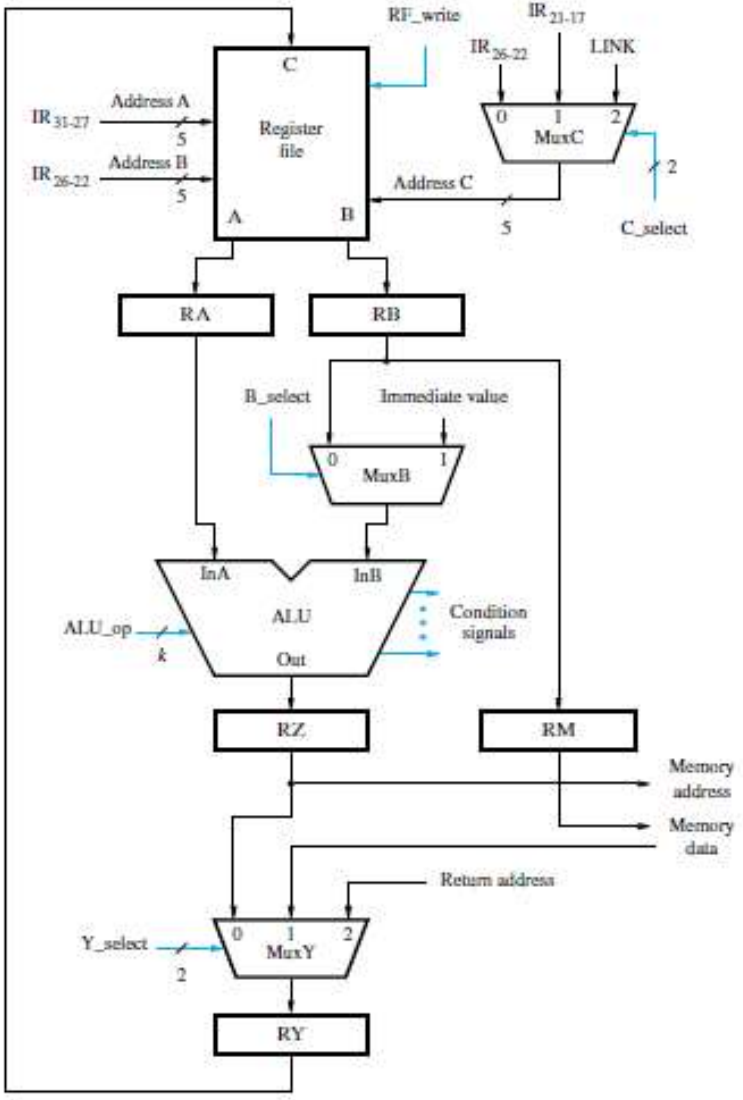


Figure 5.18 Control signals for the datapath.

Data Path Control Signals (2)

- Datapath Control Signals
  - Inter-stage registers RA, RB, RZ, RM, and RY are always enabled. This means that data flow automatically from one datapath stage to the next on every active edge of the clock signal.
  - The desired setting of various control signals can be determined by examining the actions taken in each execution step of every instruction.
    - RF_write = T5 · (ALU + Load + Call)

Data Path Control Signals (3)

- Datapath Control Signals (ctd.)
  - Recall: Add R3, R4, R5 (ctd.)
    - Sequence of Actions

Step	Action
1	Memory address ← [PC], Read memory, IR ← Memory data, PC ← [PC] + 4
2	Decode instruction, RA ← [R4], RB ← [R5]
3	RZ ← [RA] + [RB]
4	RY ← [RZ]
5	R3 ← [RY]

Figure 5.11 Sequence of actions needed to fetch and execute the instruction: Add R3, R4, R5.

Data Path Control Signals (4)

- Datapath Control Signals (ctd.)
  - Recall: Load R5, X(R7)
    - Sequence of Actions

Step	Action
1	Memory address ← [PC], Read memory, IR ← Memory data, PC ← [PC] + 4
2	Decode instruction, RA ← [R7]
3	RZ ← [RA] + Immediate value X
4	Memory address ← [RZ], Read memory, RY ← Memory data
5	R5 ← [RY]

Figure 5.13 Sequence of actions needed to fetch and execute the instruction: Load R5, X(R7).

Data Path Control Signals (5)

- Datapath Control Signals (ctd.)
  - Example: Call_Register R9

Step	Action
1	Memory address ← [PC], Read memory, IR ← Memory data, PC ← [PC] + 4
2	Decode instruction, RA ← [R9]
3	PC-Temp ← [PC], PC ← [RA]
4	RY ← [PC-Temp]
5	Register LINK ← [RY]

Figure 5.17 Sequence of actions needed to fetch and execute the instruction: Call_Register R9.

Advantage & Disadvantages

- Hardwired control provides highest speed.
- This greater speed coupled with the much higher cost of the hardwired systems tended to restrict their use to high performance computers.
- Inflexible.
- RISCs are implemented with hardwired control.
- With the trend toward simpler instructions and control, and the advent of computer aided design (CAD) tools, the design of hardwired control unit has become much easier and less prone to errors.
- If the instruction set becomes very complex (CISCs) implementing hardwired control is very difficult. In this case microprogrammed control units are used.

Summary

- 知识点：Hardwired Control Unit
- 掌握程度
  - 会画课本图5.21，并描述原理。

Content of this lecture

- 5.5 Control Signals
- 5.6 Hardwired Control
- 5.7.2 Microprogrammed Control

Microprogrammed Control (1)

- It is possible to use a “software" approach, in which the desired setting of the control signals in each step is determined by a program stored in a special memory.
- Control Word (Microinstruction)
  - Suppose that *n* control signals are needed. Let each control signal be represented by a bit in an *n*-bit word, which is often referred to as a *control word* or a *microinstruction*.
- Microroutine (Microprogram)
  - A sequence of CWs corresponding to the control sequence of a machine instruction constitute the *microroutine* for that instruction.

Microprogrammed Control (2)

- Microprogram Memory or Control Store
  - The microprogram is stored on the processor chip in a small and fast memory called the *microprogram memory* or the *control store*.
- A processor instruction is implemented by a sequence of microinstructions that are placed in a control store.

Microprogrammed Control (3)

- Microprogrammed Control Unit

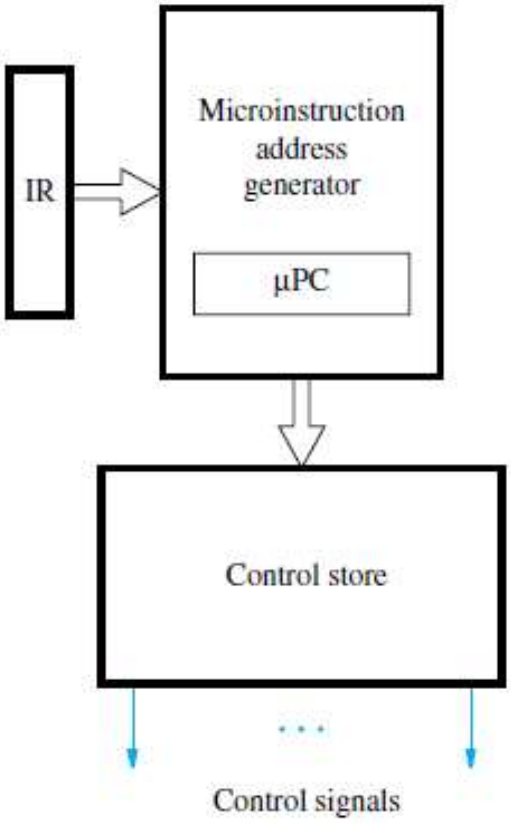


Figure 5.27 Microprogrammed control unit organization.

Microprogrammed Control (4)

- Microinstruction Encoding Schemes
  - Scheme1: Assigning one bit position to each control signal.
    - Drawbacks
      - Assigning individual bits to each control signal results in long microinstructions.
      - Only a few bits are set to 1 in any given microinstructions, so the available bit space is poorly used.

Microprogrammed Control (5)

- Microinstruction Encoding Schemes (ctd.)
  - Scheme2: Grouping signals into fields
    - Signals can be grouped so that all mutually exclusive signals are placed in the same group. Further natural groupings can be made for the remaining signals.
    - Advantage
      - Reduce the number of bits in each microinstruction.
    - Drawback
      - Require a little more decoding circuits.

Microprogrammed Control (6)

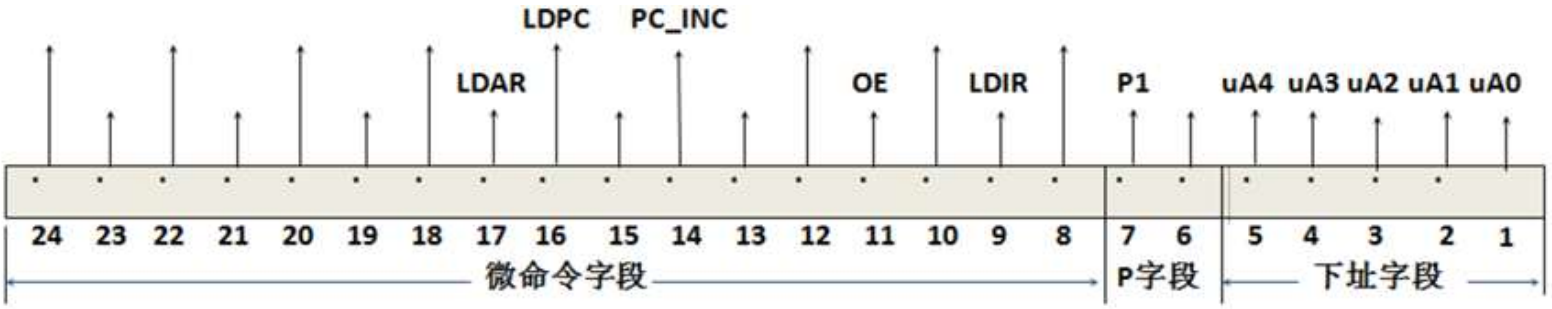
- Microprogram Sequencing
  - uPC
    - A microroutine is entered by decoding the machine instruction into a starting address that is loaded into the uPC.
    - In most cases, the microinstructions are executed sequentially.
    - Some branching capability within the microprogram can be introduced through special branch microinstructions that specify the branch address.
    - Advantage: Standard software technique can be used in writing microprograms.
    - Disadvantage: It takes more time to carry out the required branch, so the execution time is longer.

Microprogrammed Control (7)

- Microprogram Sequencing (ctd.)
  - Microinstructions with Next-address Field
    - Include an address field as a part of every microinstruction to indicate the location of the next microinstruction to be fetched.

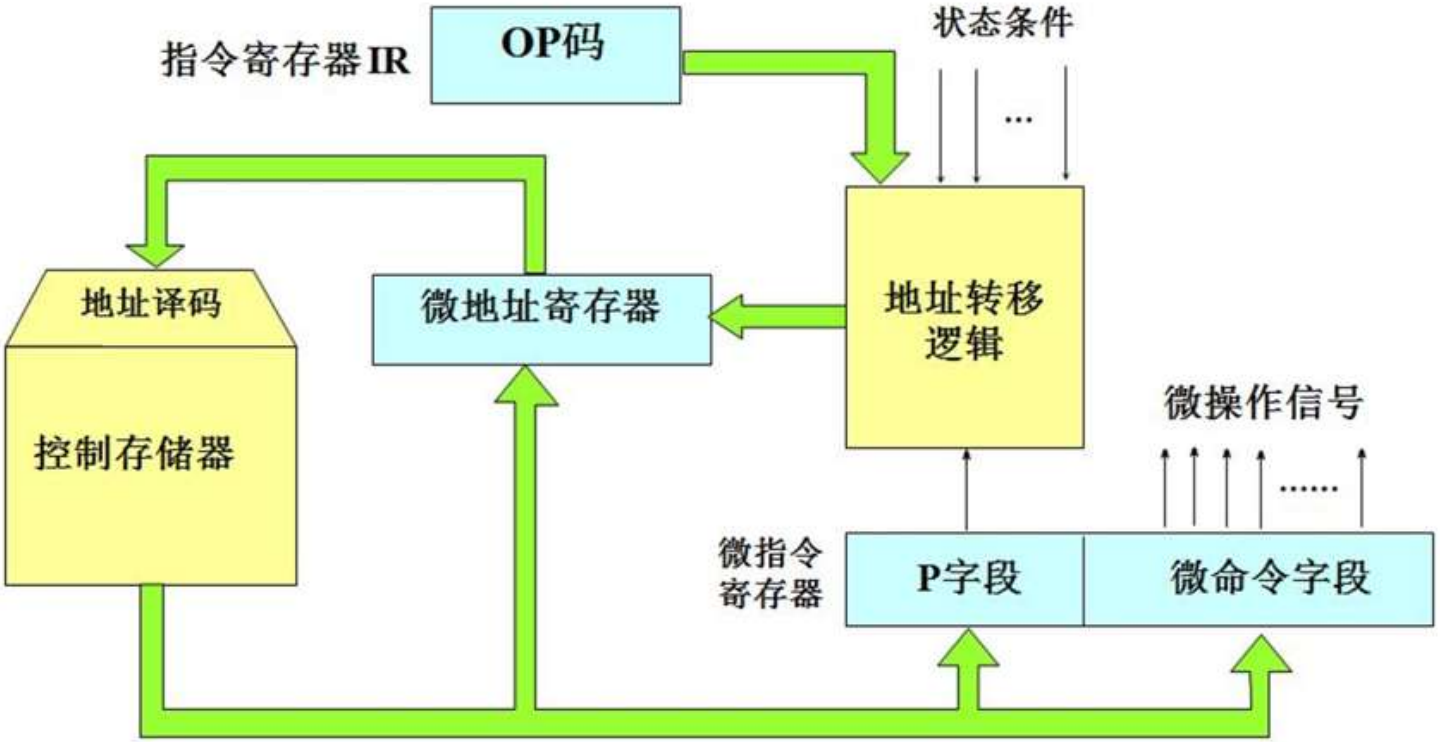
Microprogrammed Control (8)

- Microprogram Control Example (Lab4)
  - Microinstruction Format
    - 微指令字长 24 位，通过下址转移方式确定后续运行的微指令，即微指令的 1-5 位表示下一条微指令地址 [uA4, uA0]。
    - 微指令的 6-7 位留给判断字段 Px（其中 P2 位空缺），P1=1，表示本微指令是取指周期的微指令；P1=0，表示本微指令是执行周期的微指令。
    - 微指令的 8-24 位则是微命令字段，微命令即是数据通路中电平触发的微操作信号。某位置“1”，表示该位微操作信号有效；反之，置“0”则表示该位微操作信号无效。



Microprogrammed Control (9)

- Microprogram Control Example (ctd.)
  - Microprogrammed Control Unit



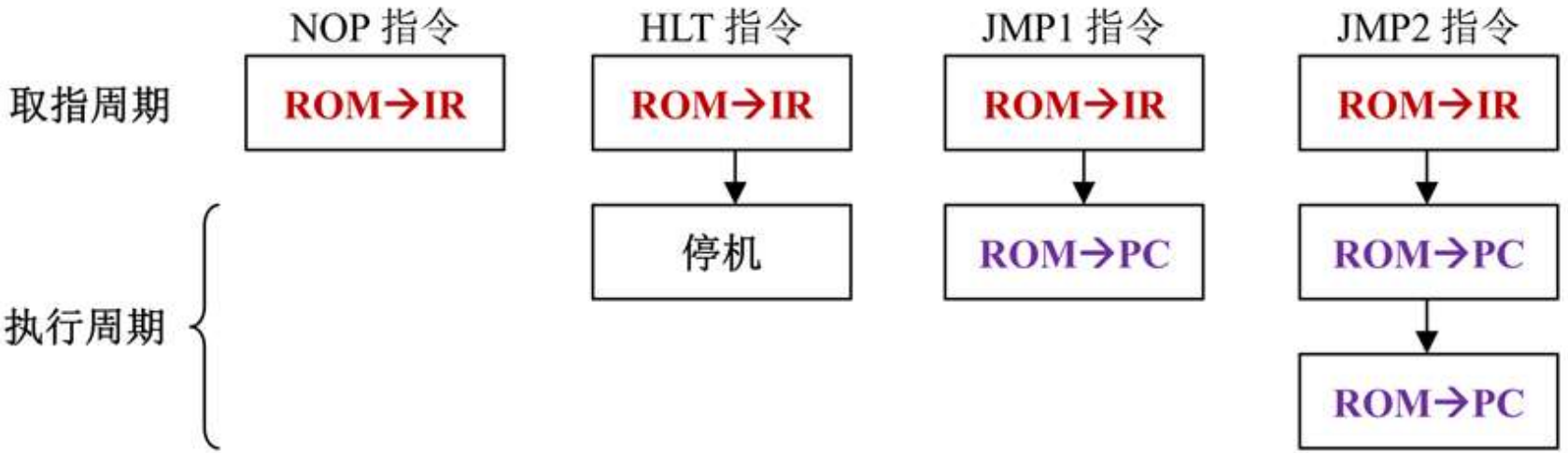
Microprogrammed Control (10)

- Microprogram Control Example (ctd.)
  - Machine Instructions

指令	OP 码 $I_7\ I_6\ I_5$	机器语言程序示例	指令功能说明
NOP	0 0 0	00000000; NOP	“空”指令，不执行任何操作
HLT	1 1 1	11100000; HLT	硬件停机
JMP1	0 0 1	00100000; JMP1 xxxxxxxx; addr1	直接寻址：程序跳转一次到地址 addr1 执行 addr1→PC;
JMP2	0 1 0	01000000; JMP2 xxxxxxxx; addr1	间接寻址：程序跳转二次到地址 addr2 执行 [addr1] = addr2, addr2 →PC;

Microprogrammed Control (11)

- Microprogram Control Example (ctd.)
  - 指令周期示意图



- 指令流（ROM->IR）是从存储器 PROGRAM 取出指令，经过总线 BUS 流向指令寄存器 IR。
- 数据流（ROM->PC）是从存储器 PROGRAM 取出数据，经过总线 BUS 流向程序计数器 PC。

Microprogrammed Control (12)

- Microprogram Control Example (ctd.)
  - 指令流/数据流的微操作信号列表

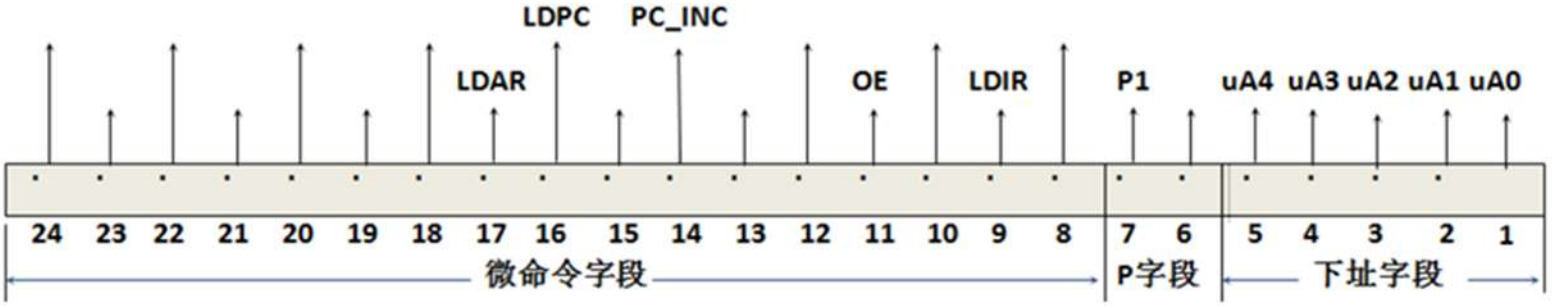
		有效的微操作信号	功能
指令流 ROM→IR	T1	$\overline{OE}$，AR_CLK（LDAR）	PC→AR, ROM→BUS
	T2	$\overline{OE}$, IR_CLK (LDIR), PC_CLK (PC_INC)	BUS→IR, PC+1
数据流 ROM→PC	T1	$\overline{OE}$, $\overline{LDPC}$, AR_CLK (LDAR)	PC→AR, ROM→BUS
	T2	$\overline{OE}$, $\overline{LDPC}$, PC_CLK (PC_INC)	BUS→PC

    - 无论是指令流还是数据流，信息都是先从一个部件打到总线 BUS，再从总线 BUS打到另一个部件的过程。
    - 所以为了保证上述操作的先后次序，指令流和数据流内部都可以分为两个周期 T1 和 T2。

Microprogrammed Control (13)

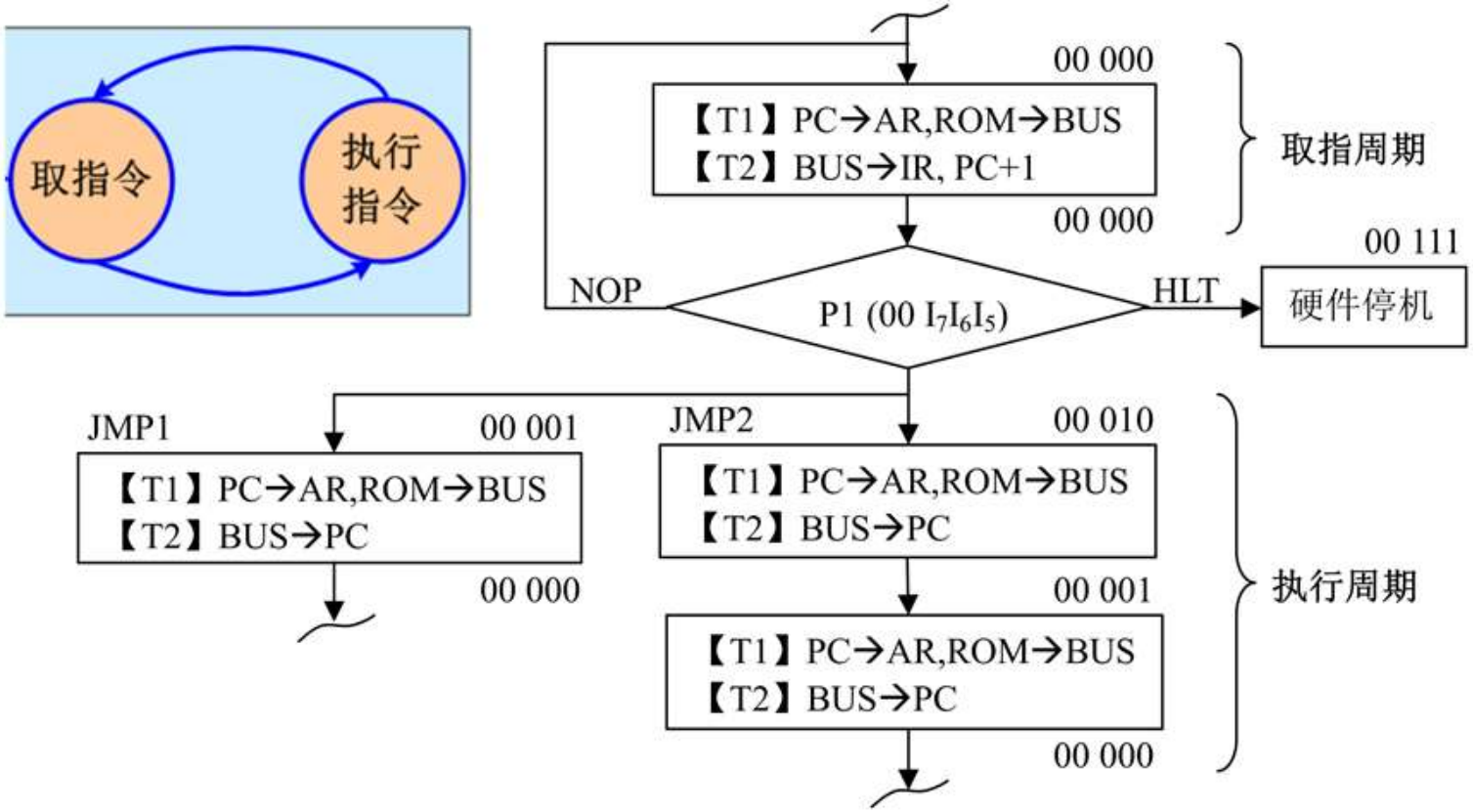
- Microprogram Control Example (ctd.)
  - 微指令代码表

Addr	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
00000	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0
00001	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
00010	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1
00111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Microprogrammed Control (14)

- Microprogram Control Example (ctd.)
  - 微程序流程图



Benefits of Microprogramming

- The decoders and sequencing logic unit of a micro-programmed control unit are very simple pieces of logic, compared to the hardwired control unit.
- Simpler design means the control unit is cheaper and less error-prone to implement.
- It is also flexible as changes could be easily made to the design.

Summary

- 知识点：Microprogram Control Unit
  - Principle of microprogram control: Figure 5.27
  - Terminologies
    - Control word : A *control word* is a word whose individual bits represent the various
    - Microprogram: A sequence of CWs corresponding to the control sequence of a machine instruction constitute the *microroutine* for that instruction.
    - Control store: The microinstruction for all instructions in the instruction set of a computer are stored in a special memory called *control store*.