جامعة ألأنبار

كلية علوم الحاسوب وتكنولوجيا المعلومات

قسم أنظمة شبكات الحاسوب

المرحلة الثانيه

Computer Architecture

التدريسي: أ.م.د عمر منذر حسين

4- A DISCUSSION ON ASSEMBLERS

- Mnemonic instructions, such as LOAD 104, are easy for humans to write and understand.
- They are impossible for computers to understand.
- Assemblers translate instructions that are comprehensible to humans into the machine language that is comprehensible to computers – We note the distinction between an assembler and a compiler:
- In assembly language, there is a one-to-one correspondence between a mnemonic instruction and its machine code.
- With compilers, this is not usually the case.
- Assemblers create an object program file from mnemonic source code in two passes.
- **During the first pass**, the assembler assembles as much of the program is it can, while it builds a symbol table that contains memory references for all symbols in the program.
- **During the second pass**, the instructions are completed using the values from the symbol table.
- Consider our example program (top).
- Note that we have included two directives HEX and DEC that specify the radix of the constants.

Address		Instruction	
	100	Load	X
	101	Add	Y
	102	Store	Z
	103	Halt	
X,	104	0023	
Y,	105	FFE9	
Z,	106	0000	

- During the first pass, we have a symbol table and the partial instructions shown at the bottom.
- After the second pass, the assembly is complete.

1 1 0 4
3 1 0 5
2 1 0 6
7 0 0 0
0 0 2 3
F F E 9
0 0 0 0

Address		Instruction	
	100	Load	X
	101	Add	Y
	102	Store	Z
	103	Halt	
X,	104	DEC	35
Y,	105	DEC	-23
Z,	106	HEX	0000

5- A DISCUSSION ON DECODING: HARDWIRED VERSUS MICROPROGRAMMED CONTROL

- The control unit, driven by the processor's clock, is **responsible for decoding the binary value in the instruction register and creating all**

necessary control signals; essentially, the control unit takes the CPU through a sequence of "control" steps for each program instruction.

- More simply put, there must be control signals to assert the control lines on the various digital components in the system, such as the ALU and memory, to make anything happen.
- Each control step results in the control unit creating a set of signals (called a control word) that executes the appropriate microoperation.
- We can opt for one of two methods to ensure that all control lines are set properly.
- **The first approach, hardwired control**, physically connects the control lines to the actual machine instructions.
- The second approach, **microprogrammed control**, employs software consisting of microinstructions that carry out an instruction's microoperations.
- **Hardwired control is very fast, but the circuits required to do this are often very complex and difficult to design, and changes to the instruction set can result in costly updates to hardwired control.**
- **Microprogrammed control is much more flexible and allows easier and less costly updates to hardware, because the program simply needs to be updated; however, it is typically slower than hardwired control.**

5.1 Hardwired Control

- Hardwired control uses the bits in the instruction register to generate control signals by feeding these bits into fixed combinational logic gates.
- There are three essential components common to all hardwired control units:
- the **instruction decoder, the cycle counter,** and the **control matrix**.
- Depending on the complexity of the system, specialized registers and sets of status flags may be provided as well.
- Figure below illustrates a simplified control unit.

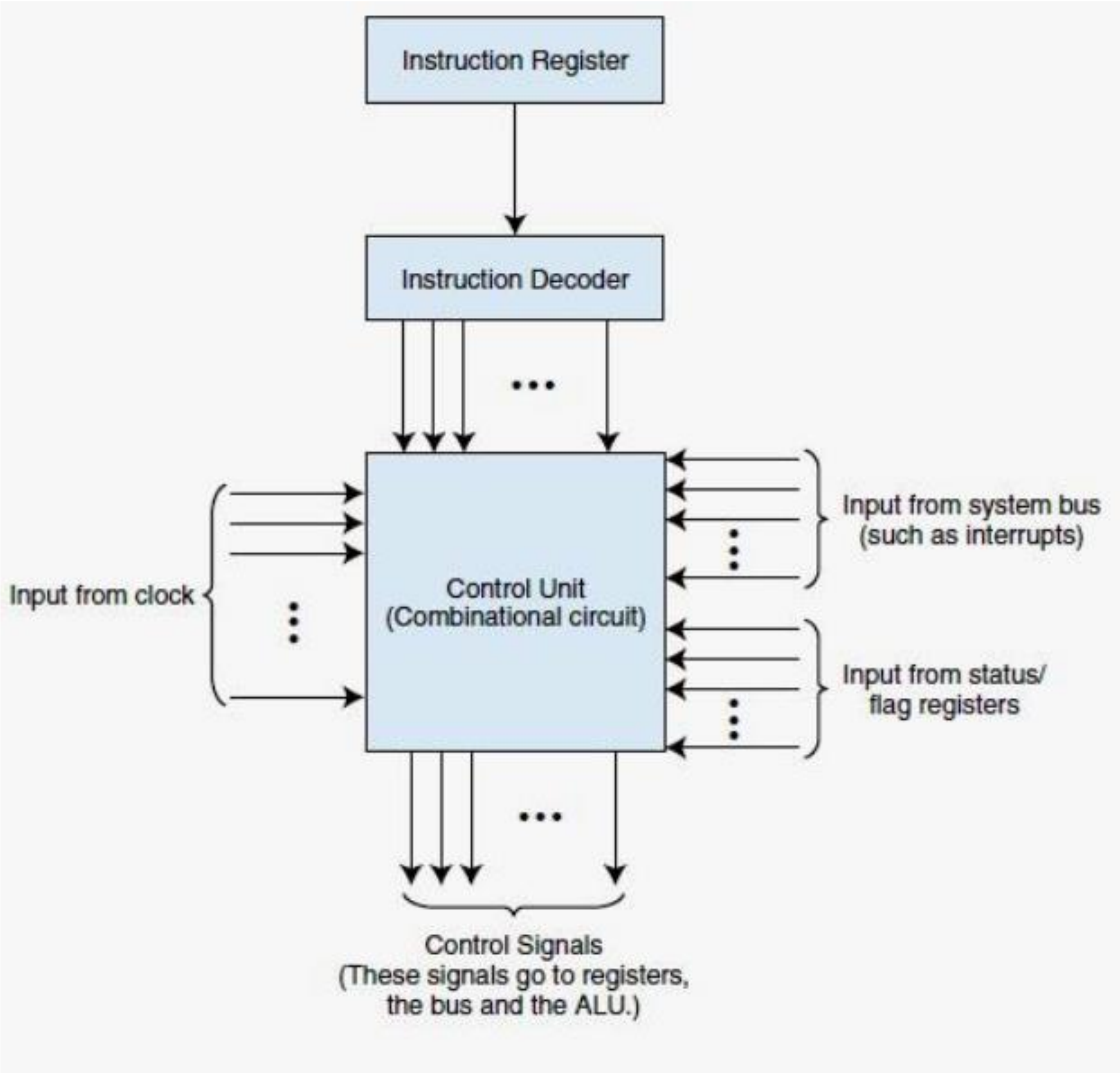

Figure 3.5 The hardwired control unit.

- The first essential component is the instruction decoder. Its job is to raise the unique output signal corresponding to the opcode in the instruction register. If we have a 4-bit opcode, the instruction decoder could have as many as 16 output signal lines.
- The advantage of hardwired control is that it is very fast. The disadvantage is that the instruction set and the control logic are tied

together directly by complex circuits that are difficult to design and modify.

- If someone designs a hardwired computer and later decides to extend the instruction set, the physical components in the computer must be changed. This is prohibitively expensive, because not only must new chips be fabricated, but the old ones must be located and replaced.

5.2 Microprogrammed Control

- For a computer with a large instruction set, it might be virtually impossible to implement hardwired control.
- In microprogrammed control, instruction microcode produces the necessary control signals.
- A generic block diagram of a microprogrammed control unit is shown in Figure below.

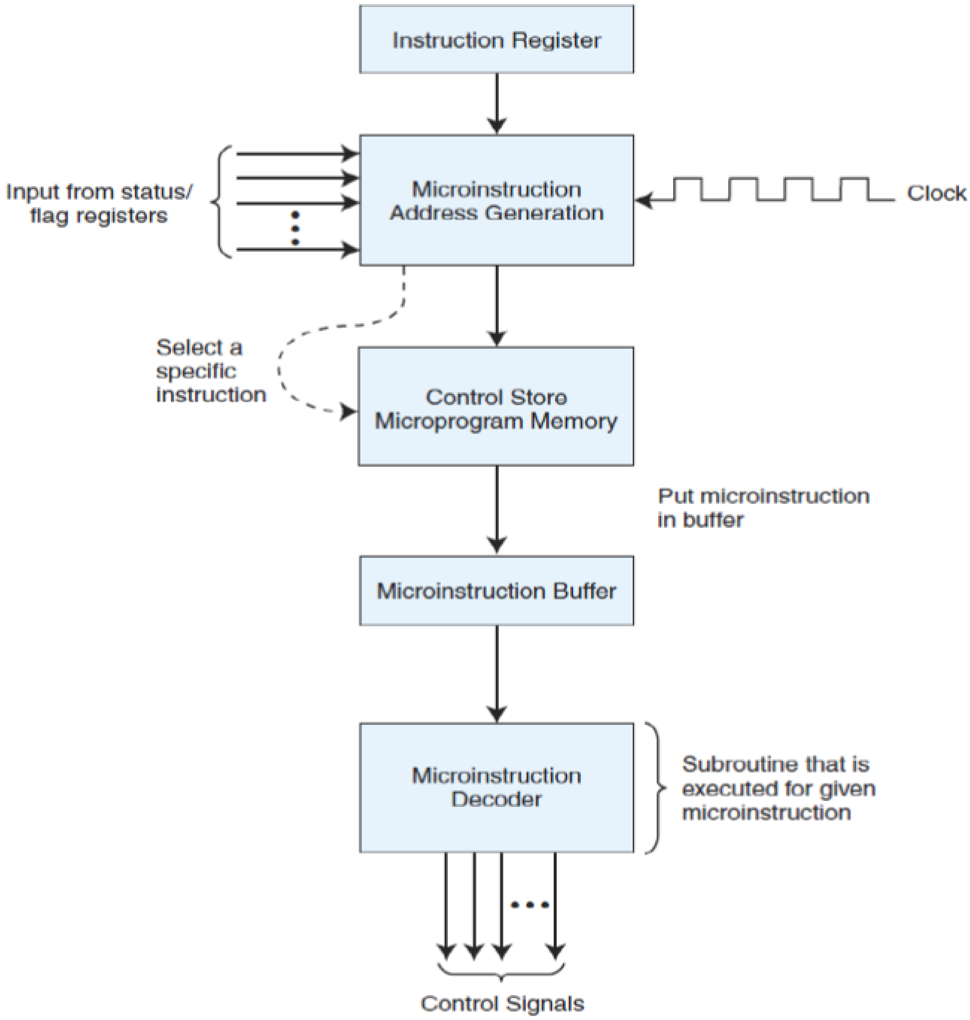

Figure 3.6 The microprogramed control unit.

- All machine instructions are input into a special program, the microprogram, that converts machine instructions of 0s and 1s into control signals.
- The microprogram is essentially an interpreter, written in microcode, that is stored in firmware (ROM, PROM, or EPROM), which is often referred to as the control store.
- The great advantage of microprogrammed control is that if the instruction set requires modification, only the microprogram needs to be updated to match: No changes to the hardware are required.

- Thus, microprogrammed control units are less costly to manufacture and maintain.
- Because cost is critical in consumer products, microprogrammed control dominates the personal computer market.

6 REAL-WORLD EXAMPLES OF COMPUTER ARCHITECTURES

- MARIE shares many features with modern architectures, but itis not an accurate depiction of them.
- Two contemporary computer architectures better illustrate the features of modern architectures.
- The Intel architecture (the x86 and the Pentium families) and then follow with the MIPS architecture.
- Each member of the x86 family of Intel architectures is known as a CISC (complex instruction set computer) machine, whereas the Pentium family and the MIPS architectures are examples of RISC (reduced instruction set computer) machines.
- CISC machines have a large number of instructions, of variable length, with complex layouts.
- Many of these instructions are quite complicated, performing multiple operations when a single instruction is executed (e.g., it is possible to do loops using a single assembly language instruction).
- The basic problem with CISC machines is that a small subset of complex CISC instructions slows the systems down considerably.
- Designers decided to return to a less complicated architecture and to hardwire a small (but complete) instruction set that would execute extremely quickly.
- This meant it would be the compiler’s responsibility to produce efficient code for the ISA.
- Machines utilizing this philosophy are called RISC machines.
- In RISC the number of instructions is reduced.

- However, the main objective of RISC machines is to simplify instructions so they can execute more quickly.
- Each instruction performs only one operation, they are all the same size, they have only a few different layouts, and all arithmetic operations must be performed between registers (data in memory cannot be used as operands).
- Virtually all new instruction sets (for any architectures) since 1982 have been RISC, or some sort of combination of CISC and RISC.