Computer Organization and Architecture

Chapter: Control Unit Design Issue

Exercise & Exam-Oriented Questions with Answers

This set covers the most frequently asked short-answer and broad (long-answer) questions from the chapter Control Unit Design Issue, along with complete answers. Topics covered include hardwired control unit design, microprogrammed control unit design, horizontal and vertical microprogramming, and a comparison between the two design approaches. Use this set for quick revision as well as for practising full exam-style answers.

Section A — Short Answer Questions

Q. What is a control unit?

Ans. The control unit is a component of the CPU that generates the timing and control signals required to fetch, decode, and execute instructions. It coordinates the operation of the ALU, registers, memory, and I/O devices during instruction execution.

Q. What are the two basic approaches to designing a control unit?

Ans. The two basic approaches are: (i) Hardwired control unit design, in which control signals are generated using fixed logic circuits, and (ii) Microprogrammed control unit design, in which control signals are generated by a stored microprogram held in control memory.

Q. Define a hardwired control unit.

Ans. A hardwired control unit is one in which control signals are generated by fixed combinational and sequential logic circuits (gates, flip-flops, decoders) that are physically wired to produce the required outputs; the design cannot be changed without altering the hardware.

Q. Define a microprogrammed control unit.

Ans. A microprogrammed control unit generates control signals using a microprogram — a sequence of microinstructions — stored in a control memory (control store). Each machine instruction corresponds to a microroutine made up of one or more microinstructions.

Q. What is control memory (control store)?

Ans. Control memory is a special, high-speed memory (usually ROM, sometimes writable) inside a microprogrammed control unit that stores all the microinstructions needed to generate control signals for every machine instruction in the instruction set.

Q. What is the function of the Control Address Register (CAR)?

Ans. The CAR holds the address of the next microinstruction to be fetched from control memory. It is updated after every micro-step by the sequencing logic to point to the correct next microinstruction.

Q. What is the function of the Control Buffer Register (CBR)?

Ans. The CBR (also called the microinstruction register) holds the microinstruction that is currently being executed. The bits of the CBR are used, directly or after decoding, to generate the actual control signals for that clock cycle.

Q. What is a microinstruction?

Ans. A microinstruction is a single word stored in control memory that specifies one or more micro-operations to be performed in one clock cycle, along with (in most designs) the address or address information for the next microinstruction.

Q. What is a microprogram?

Ans. A microprogram is a sequence of microinstructions stored in control memory that together implement the execution of one machine-level instruction. It is sometimes also called a microroutine.

Q. Define horizontal microprogramming.

Ans. Horizontal microprogramming is a microinstruction format in which each bit of the microinstruction directly controls one control signal or micro-operation, with little or no encoding. It allows a high degree of parallelism but results in a long microinstruction word.

Q. Define vertical microprogramming.

Ans. Vertical microprogramming is a microinstruction format in which control signals are encoded into compact fields (similar to an opcode), requiring a decoder to generate the actual control signals. It produces a shorter microinstruction word but supports less parallelism.

Q. Which control unit design is faster — hardwired or microprogrammed — and why?

Ans. The hardwired control unit is faster, because its control signals are generated directly by fixed logic circuits with no memory access involved, whereas a microprogrammed control unit must read a microinstruction from control memory for every micro-step, adding an access delay.

Q. Which type of processor typically uses a hardwired control unit, and which uses a microprogrammed control unit?

Ans. RISC (Reduced Instruction Set Computer) processors, with their small and simple instruction sets, typically use hardwired control units. CISC (Complex Instruction Set Computer) processors, with their large and complex instruction sets, typically use microprogrammed control units.

Q. State one advantage and one disadvantage of a hardwired control unit.

Ans. Advantage: it operates at very high speed since no memory access is needed to generate control signals. Disadvantage: it is inflexible — any change to the instruction set requires redesigning the physical circuit/wiring.

Q. State one advantage and one disadvantage of a microprogrammed control unit.

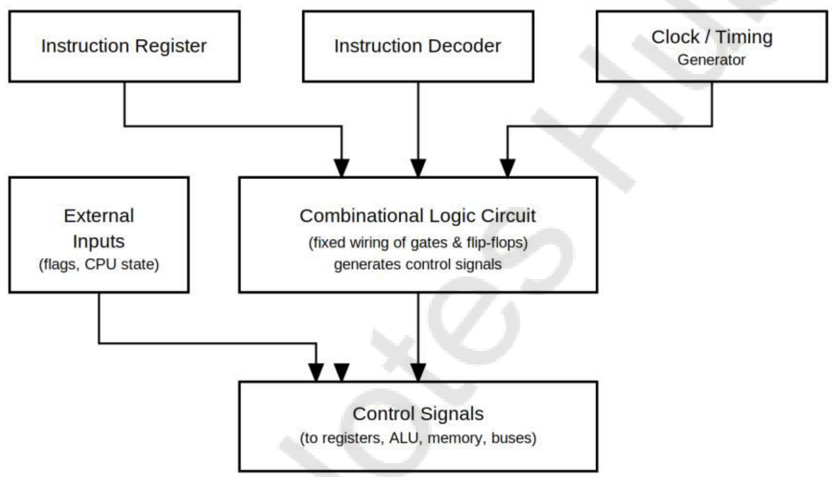
Ans. Advantage: it is highly flexible, since the instruction set can be changed or extended simply by rewriting the microprogram in control memory. Disadvantage: it is slower than hardwired control, because every micro-operation requires a control memory access.

Section B — Broad / Long Answer Questions

Q. Explain, with a neat block diagram, the working of a hardwired control unit.

Ans. A hardwired control unit generates control signals using fixed combinational and sequential logic circuits built from gates, flip-flops, decoders, and counters. Because the logic is physically wired, the design cannot be modified without redesigning the hardware itself.

Working: The instruction fetched from memory is placed in the Instruction Register (IR). The instruction decoder decodes the opcode part of the instruction. This decoded output, together with timing signals from the clock/timing generator and status flags (such as carry, zero, overflow) from the CPU, is fed into a combinational logic circuit. This circuit generates the exact set of control signals required at that instant, which are then used to drive the registers, ALU operations, memory read/write lines, and internal buses.



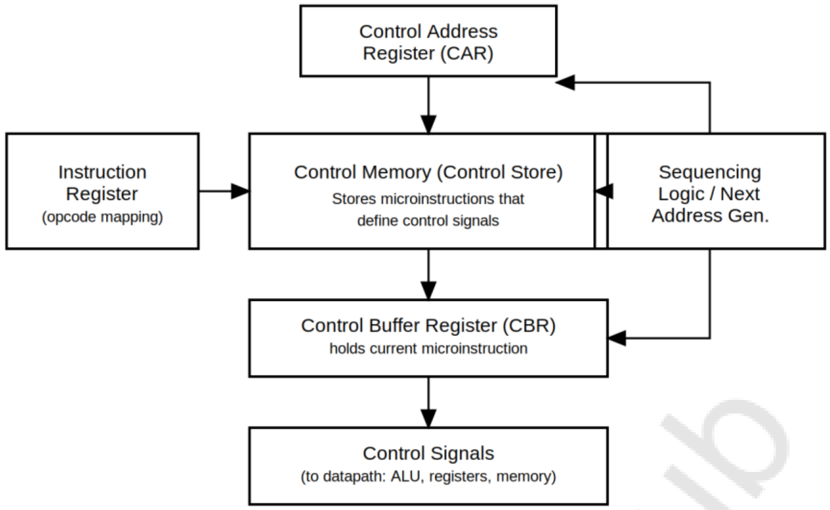
Block diagram of a hardwired control unit

Because control signals are produced directly by logic gates without any memory access, this design is very fast, but it is difficult to modify and is best suited to processors with a small, fixed instruction set, such as RISC processors.

Q. Explain, with a neat block diagram, the working of a microprogrammed control unit.

Ans. A microprogrammed control unit generates control signals using a stored program called a microprogram, held in a special memory called control memory (control store). Each machine instruction is implemented by a microroutine — a sequence of microinstructions — and each microinstruction specifies the micro-operations for one clock cycle.

Working: The opcode of the machine instruction is first mapped to the starting address of its microroutine in control memory. This address is loaded into the Control Address Register (CAR). The microinstruction stored at that address is then fetched into the Control Buffer Register (CBR), and its bits generate the control signals for that micro-step, either directly or after decoding. Meanwhile, the sequencing logic (next address generator) computes the address of the next microinstruction, based on the current microinstruction, status flags, and any conditional branching within the microroutine, and this address is loaded back into the CAR. The cycle repeats until the entire microroutine for the instruction has been executed.



Block diagram of a microprogrammed control unit

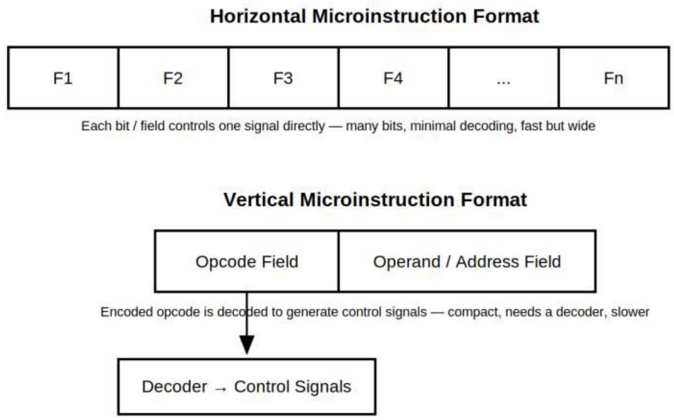
This approach is slower than hardwired control because of the repeated control memory accesses, but it is far more flexible, since the instruction set can be changed simply by editing the microprogram rather than redesigning hardware. It is well suited to CISC processors with large, complex instruction sets.

Q. What is a microinstruction? Explain the concepts of horizontal and vertical microprogramming with suitable diagrams.

Ans. A microinstruction is a single control word stored in control memory that specifies the micro-operations to be carried out in one clock cycle, along with information about the next microinstruction to be executed. The way in which control information is organised within this word gives rise to two microinstruction formats: horizontal and vertical microprogramming.

Horizontal microprogramming: Each bit (or field) of the microinstruction is directly associated with one control signal, with little or no encoding. Because many bits can be set to 1 simultaneously, a large number of control lines can be activated in the same clock cycle, giving a high degree of parallelism. This results in a long microinstruction word and a larger control memory, but faster execution since no decoding is needed.

Vertical microprogramming: Control signals are encoded into compact fields, similar to an opcode, instead of being represented bit by bit. A decoder is required to interpret the encoded field and produce the actual control signals. This results in a short, compact microinstruction word and a smaller control memory, but supports less parallelism and adds a small decoding delay.



Horizontal versus vertical microinstruction format

In short, horizontal microprogramming trades a longer control word for speed and parallelism, while vertical microprogramming trades some speed and parallelism for a shorter, more compact control word.

Q. Differentiate between horizontal and vertical microinstruction formats.

Ans. The two formats differ mainly in word length, decoding requirement, and degree of parallelism, as summarised below:

Basis	**Horizontal Microprogramming**	**Vertical Microprogramming**
Word length	Long — one bit per control signal	Short — signals encoded into compact fields
Decoding	Little or none required	Decoder required to interpret encoded field
Parallelism	High — many signals active together	Lower — limited operations encoded per word
Control memory size	Larger, due to long word length	Smaller, due to compact word length
Speed	Faster (no decoding delay)	Slightly slower (decoding required)

Q. Compare hardwired control unit and microprogrammed control unit.

Ans. Both designs aim to generate the control signals needed to execute instructions, but they differ significantly in speed, flexibility, cost, and complexity:

Basis	**Hardwired Control Unit**	**Microprogrammed Control Unit**
Signal generation	Fixed logic circuits	Stored microinstructions in control memory
Speed	Faster	Slower (control memory access needed)
Flexibility	Inflexible; needs redesign to change	Flexible; change the microprogram only
Design complexity	Complex, hard to debug	Simpler, systematic design
Cost	Lower for simple instruction sets	Higher, due to control memory
Suitability	RISC processors	CISC processors

Q. Explain the advantages and disadvantages of the microprogrammed control unit over the hardwired control unit.

Ans. Advantages of the microprogrammed control unit:

- Highly flexible — the instruction set can be modified, corrected, or extended by simply rewriting the microprogram, without changing any hardware.
- Easier to design and debug, since it follows a systematic, program-like approach rather than complex ad-hoc wiring.
- Well suited to large and complex instruction sets, as required by CISC processors.
- Errors in control logic can be corrected by simply updating the control memory content (if it is writable).

Disadvantages of the microprogrammed control unit:

- Slower than hardwired control, because every micro-operation requires an access to control memory.
- Requires additional hardware in the form of control memory, increasing cost.
- For very simple, fixed instruction sets, it can be less efficient than a hardwired design.

Q. Why are hardwired control units typically used in RISC processors while microprogrammed control units are typically used in CISC processors?

Ans. RISC processors have a small, simple, and fixed instruction set with instructions of uniform format, so their control logic can be implemented efficiently and cost-effectively using fixed hardware, and the resulting hardwired control unit gives very high execution speed, which matches the RISC design goal of fast, simple instruction execution. CISC processors, on the other hand, have a large number of complex instructions with variable formats and addressing modes; implementing this variety through fixed wiring would make the hardwired design extremely complicated and difficult to modify. A microprogrammed control unit handles this complexity systematically, since each complex instruction can be implemented as a microroutine, and the instruction set can be extended or corrected simply by updating the microprogram, making it far better suited to CISC processors.

Q. Discuss the various design issues that must be considered while designing a control unit for a digital computer.

Ans. While designing a control unit, a computer architect must consider several key issues:

- Choice of design approach: whether to use a hardwired design (for speed and simplicity) or a microprogrammed design (for flexibility and support of a complex instruction set).
- Speed requirements: hardwired designs offer higher speed, while microprogrammed designs trade some speed for flexibility.
- Size and complexity of the instruction set: a large, complex instruction set favours a microprogrammed approach, while a small, uniform instruction set favours a hardwired approach.
- Flexibility and scope for future modification: whether the instruction set is likely to change or be extended, which favours microprogramming.
- Cost of implementation: hardwired designs need only logic circuits, while microprogrammed designs need additional control memory, raising the cost.
- Ease of design, testing, and debugging: microprogrammed designs are generally easier to design and verify systematically.
- Choice of microinstruction format (if microprogrammed): horizontal format for higher parallelism and speed versus vertical format for a more compact control memory.