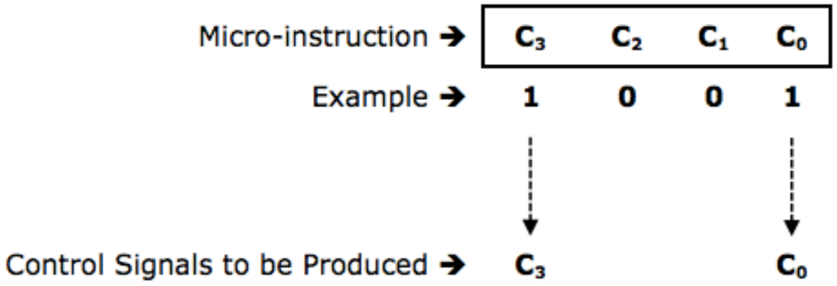
MICRO-INSTRUCTION FORMAT

The main part of the micro-instruction is its control field.
It determines the control signals to be produced.
It can be of two different formats: Horizontal or Vertical.

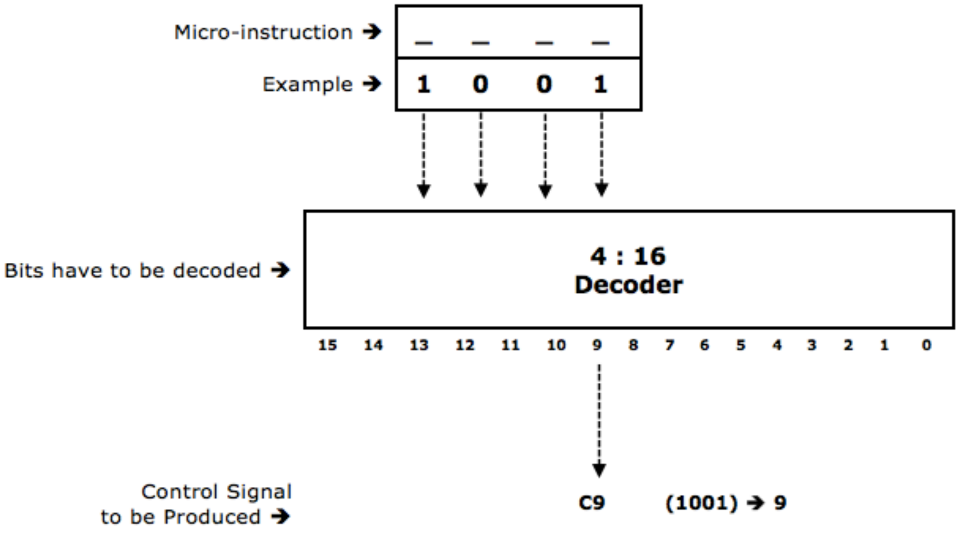
1) HORIZONTAL MICRO-INSTRUCTION

Here every bit of the micro-instruction corresponds to a control signal.
Whichever bit is "1", that particular control signal will be produced by the micro-instruction.



2) VERTICAL MICRO-INSTRUCTION

Here bits of the micro-instruction have to be decoded.
The decoded output decides the control signal to be produced.

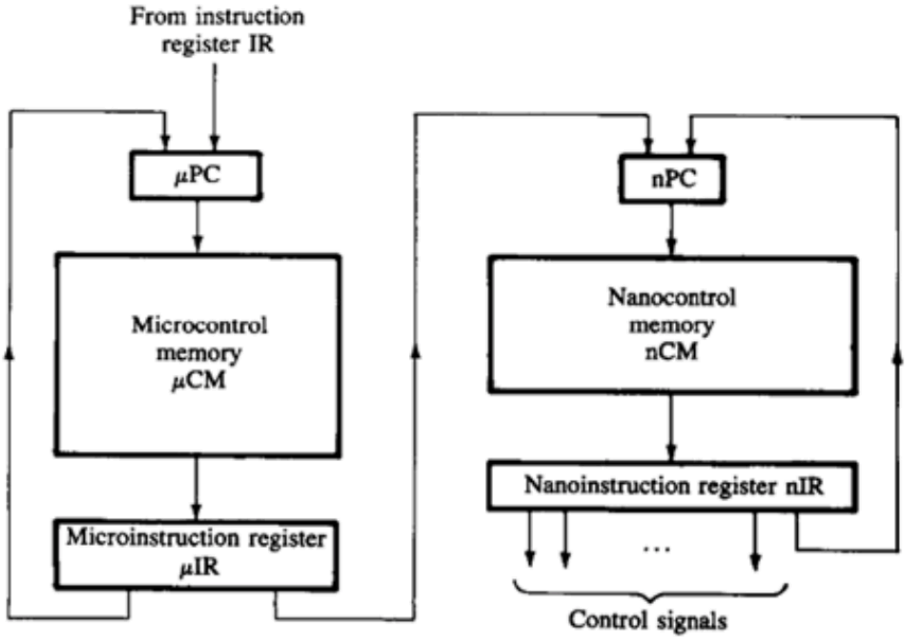


	HORIZONTAL MICRO-INSTRUCTION	VERTICAL MICRO-INSTRUCTION
1	Every bit of the micro-instruction corresponds to a control signal.	Bits of the micro-instruction have to be decoded to produce control signals.
2	Does not require a decoder.	Needs a decoder.
3	N bits in the micro-instruction will totally produce N control signals.	N bits in the micro-instruction will totally produce 2^N control signals.
4	Multiple control signals can be produced by one micro-instruction.	Only one control signal can be produced by one micro-instruction.
5	As the control signals increase, the micro-instruction grows wider. Hence the Control Memory grows Horizontally.	To produce more control signals, more number of micro-instructions are needed. Hence the Control Memory grows Vertically.
6	Executes faster as no decoding needed.	Executes slower as decoding is needed.
7	Micro-instruction are very wide. Hence Control memory is large.	Micro-instruction are much narrower. Hence Control memory is small.
8	Circuit is simpler as a decoder is not needed.	Circuit is more complex as a decoder is needed.

As seen from the above comparison, both methods have their pros and cons.
So a combination of both is used together called Nano-Programming.

NANO-PROGRAMMING *(Very Important)*

1) **Horizontal** μ-instructions can produce **multiple control signals** simultaneously, but are **very wide.**
2) This makes the Control Memory **very large in size**.
3) **Vertical** micro-instructions are **narrow**, **but on decoding** can **produce** only **one control signal**.
4) This makes the Control Memory **small** but the execution is **slow.**
5) Hence a **combination of both techniques** is needed called **Nano-Programming**.
6) Here we have a **two level control memory**.
7) The instruction is fetched from the **main memory into IR**.
8) Using its **opcode** we load **address of its first micro-instruction** into **μPC**,
9) Using this address we **fetch the micro-instruction** from μ-Control Memory (**μCM**) **into μIR**.
10) This is in **vertical form** and has to be decoded.
11) The decoded output **loads a new address** in a Nano program counter (**nPC**).
12) Using this address we **fetch the Nano-instruction** from Nano-Control Memory (**nCM**) into **nIR**.
13) This is in **horizontal form** and can **directly generate control signals**.
14) Such a combination **gives advantage of both techniques**.
15) The size of the Control Memory is **small** as μ-instructions are **Vertical.**
16) Multiple control signals can be **produced simultaneously** as Nano-instructions are **Horizontal**.


From instruction register IR
μPC
Microcontrol memory μCM
Microinstruction register μIR
nPC
Nanocontrol memory nCM
Nanoinstruction register nIR
...
Control signals

	HARDWIRED CONTROL UNIT	MICROPROGRAMMED CONTROL UNIT
1	**Control signals are generated using hardware**.	**Control signals are generated using software (Microprogram).**
2	Since **hardware** is used, the circuit is **rigid**.	Since **software** is used, the circuit is **flexible.**
3	**Modification** to the Control Unit requires **re-design** of the entire **hardware**.	**Modification** to the Control Unit simply requires **re-programming** of µ-instructions.
4	**Ideally suited** for processors with **small and simple instruction sets**.	**Ideally suited** for processors with **large and complex instruction sets**.
5	**Debugging** a large Hardwired Control Unit is **very difficult.**	As micro-programs are software, **debugging is much easier.**
6	**Emulation is not possible.**	**Emulation is possible.**
7	Executes **faster** as control signals are directly generated by hardware.	Executes **slower** as **time** is **wasted** in **fetching and decoding µ-instructions**.
8	Does not need a **Control Memory**.	Needs a **Control Memory**.
9	**Cost is lower** as Control Memory is not needed.	**Cost is higher** as Control Memory is needed inside the processor.
10	Preferred in **RISC processors**.	Preferred in **CISC processors**.

As seen from the above comparison, both methods have their pros and cons.
Hence modern processors use a combination of both.
Simple and regularly used instructions are decoded by a Hardwired Control Unit as they are faster.
Complex instructions are decoded by a Microprogrammed Control Unit as they are easier to design..