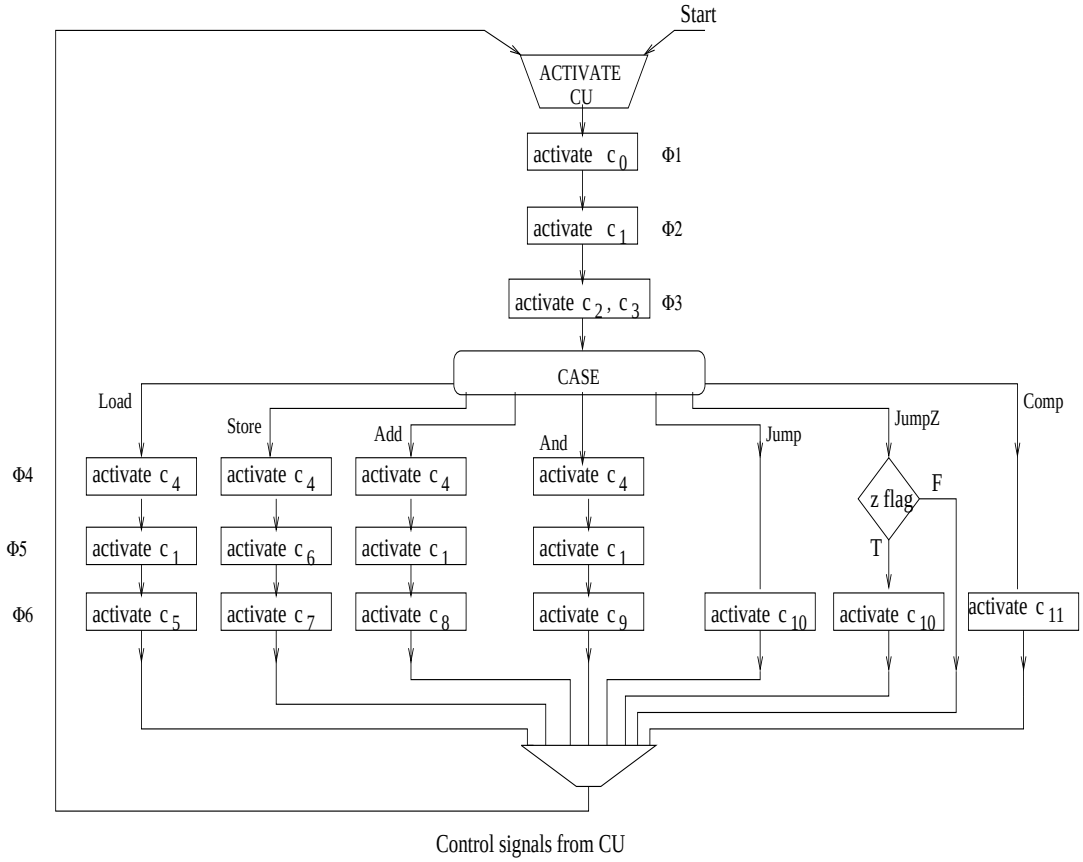
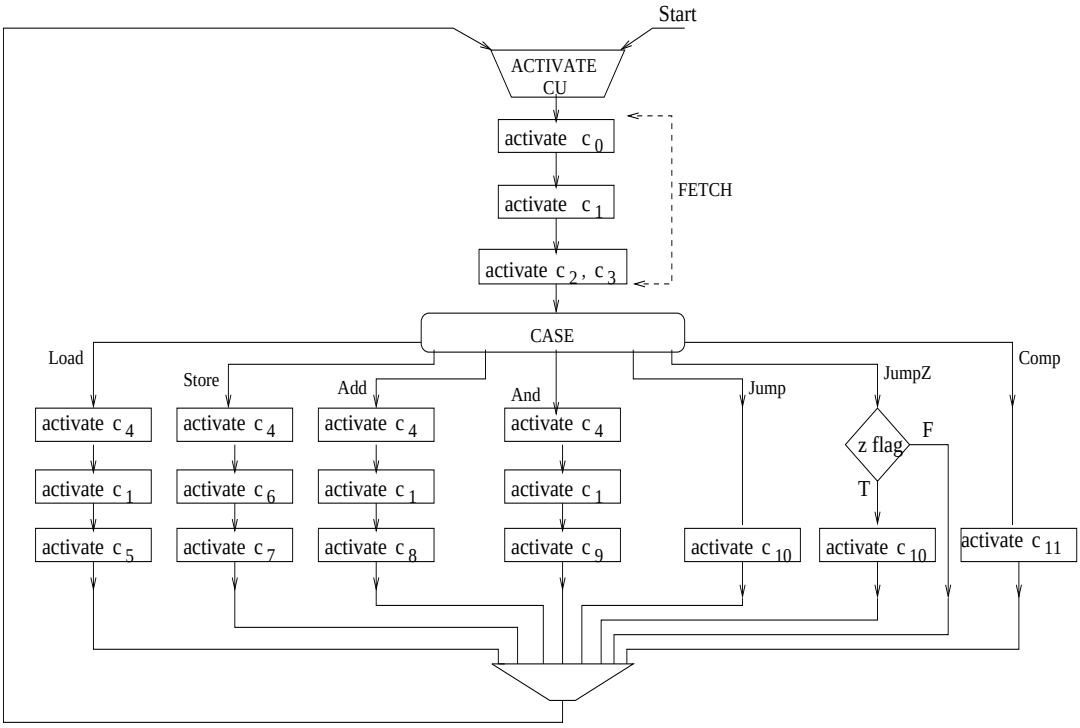
Lecture 15: March 16, 2021
Computer Architecture and Organization-I
Biplab K Sikdar

0.3 Microprogrammed Control Design

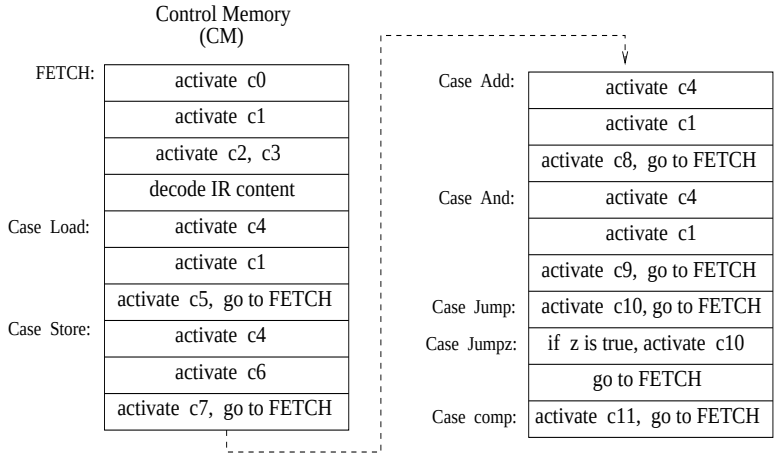
Target is to generate the sequence of control signals as noted in following figure.



In a microprommed control design this can be achieved by executing a set of microinstructions (microprogramme) as shown in Figure 11(b).



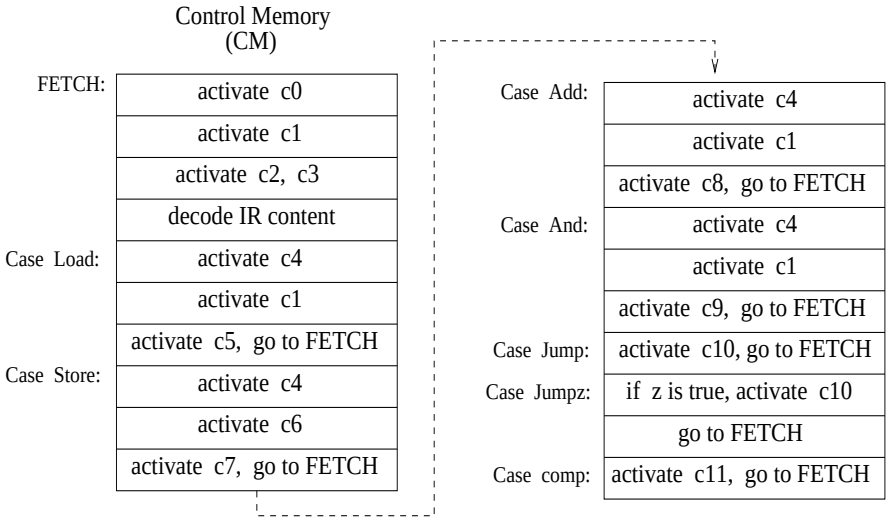
(a) Control signals from micro programmed CU



(b) Microprogram for CU design

Figure 11: Overview of micro-programmed control design

Microinstructions are stored in a special class of memory - control memory (CM).



Microprogram for CU design

Content of CM describes that the if the CPU is switched on, CU can fetch μ-instruction *activate* c_0 and execute.

It ensures tranfer of PC content to AR.

Then CU fetches and executes *activate* c_1 that realizes memory read DR $\leftarrow$ M(AR).

That is, execution of the set of microinstructions stored im CM realizes CU function.

The design technique demands attention to the following issues:

1. Encoding of μ-instructions.
2. μ-instruction sequencing.

   After activation of c_0 it is to be ensured that *activate* c_1 is fetched from CM.
3. Address mapping -once macroinstruction fetched from MM is found *Store* (say), then CU must fetch μ-instruction from address *case Store* (Figure 11(b)).

   That is, depending on content of IR the CU should decide on next executable μ-instruction address.

0.3.1 Micro-instruction encoding

CU fetches an instruction to a register called control memory data register (CMDR) or microinstruction register (μIR) (Figure 12).

Decoding of microinstruction is required to activate one (*activate* c_0) or multiple (*activate* c_2, c_3) control signals.

Outputs of the decoder (control signals) are input to different parts of CPU.

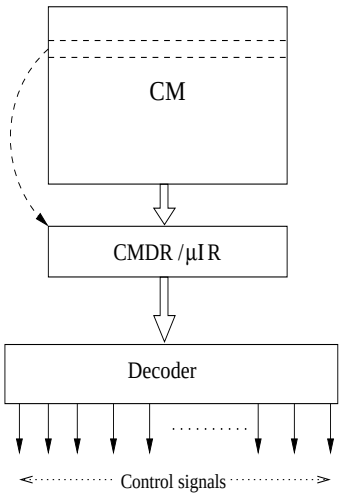


Figure 12: Microinstruction decoding

Encoding controls size of a CM.

Encoding scheme can reduce the CM word size but may increase delay in decoding μ-instructions as well as limit possibility of parallel activation of control signals.

Three encoding schemes are considered for μ-programmed CU instruction format.

1. Horizontal
2. Vertical
3. Diagonal

0.3.1.1 Horizontal format

Horizontal μ-instruction encoding for CU of Figure 11(a)) is in Figure 13(a).

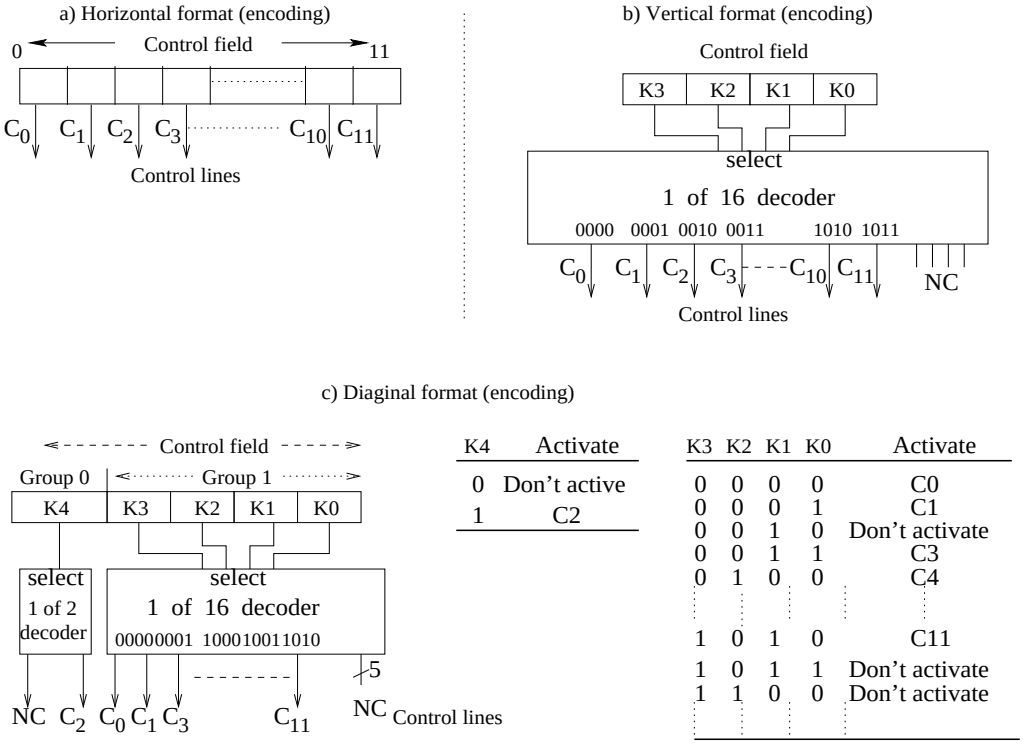


Figure 13: Microinstruction formats

It assigns one bit per control signal.

In example design, it requires 12-bit field (control field).

Horizontal format in general does not require decoder - avoids delay of decoding.

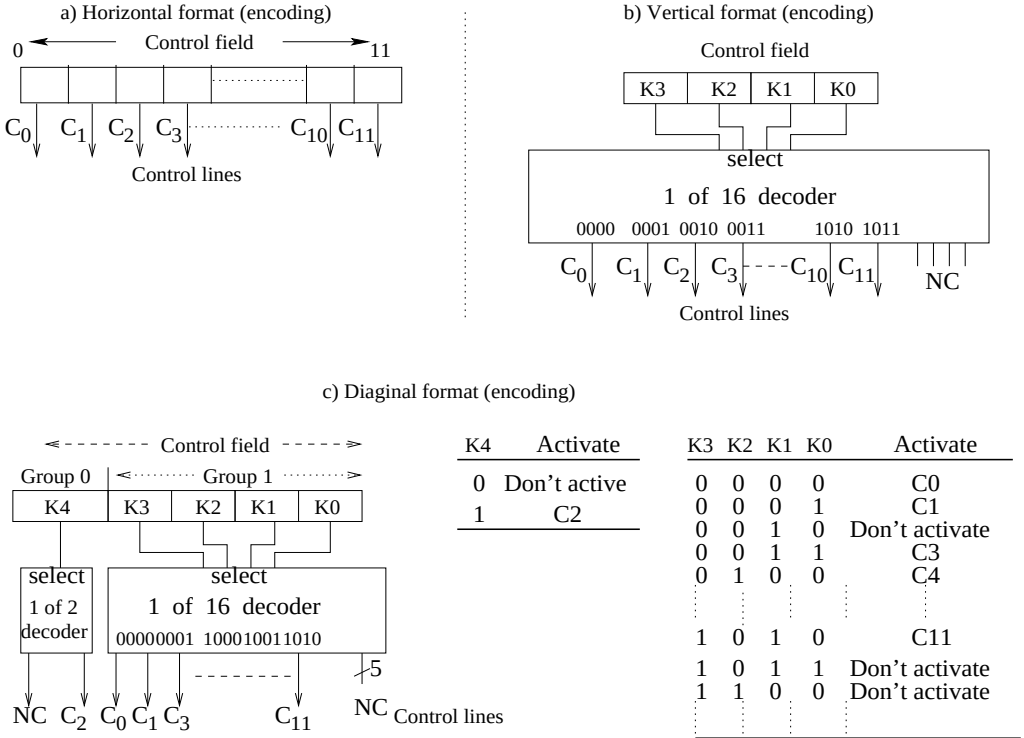
The features of horizontal format -

a) It is a long format,

b) It has ability to express high degree of parallelism,

c) It considers little encoding (generally no encoding) of control information.

0.3.1.2 Vertical format



Features of a vertical format (Figure 13(b)) are

a) This is of short format,

b) Limited ability to support parallelism (generally no parallelism) inμ-operations,

c) Accepts considerable encoding of control information.

For a CU, realized with n control signals, this demands m-bit, where $n \leq 2^m$.

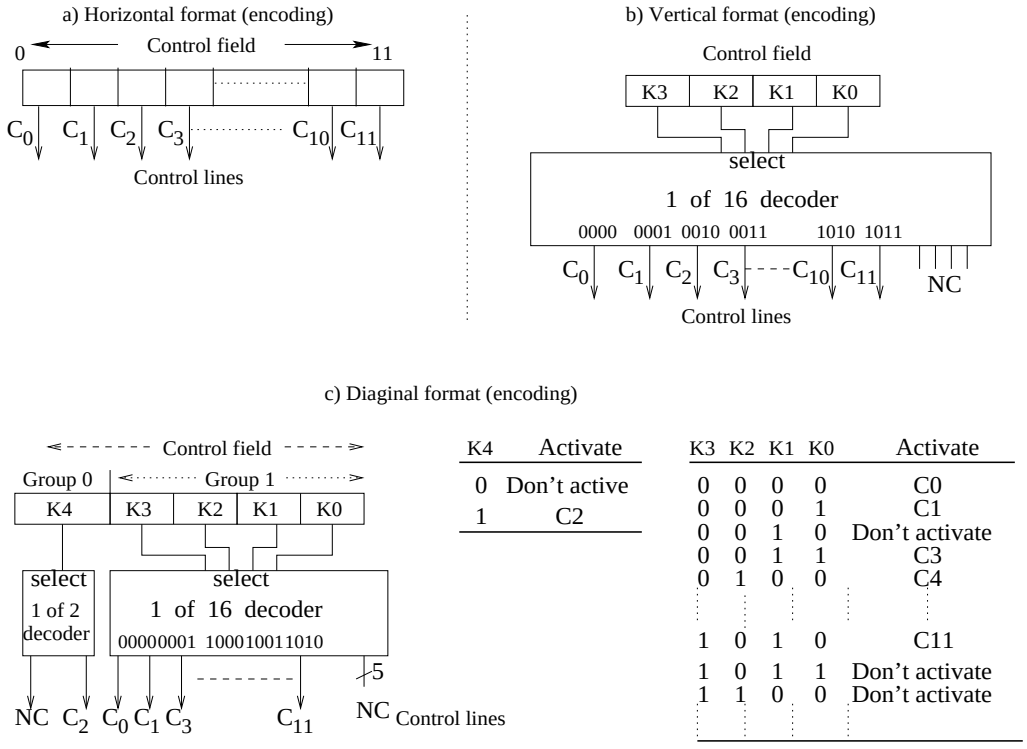
For n=12 control signals of example design, we need m=4 bits.

The 4-bit (K3 K2 K1 K0) code is then decoded by a 1-of-16 decoder.

Each output of decoder is connected to a control signal (Figure 13(b)).

For example, if K3 K2 K1 K0 = 1010, then c_{10} is activated.

0.3.1.3 Diagonal format



Limitation of vertical format - a microinstruction can't generate two control signals.

That is, parallel execution of microoperations are not allowed.

In example design, two control signals c_2 and c_3 can be considered in parallel.

Vertical encoding can't allow this. Solution is: c_2 and c_3 are generated sequentially.

Alternative solution is: n μ-instructions are partitioned into m groups to get maximum *m* parallel microoperations with in a μ-instruction.

Each group is encoded as in vertical format.

To decode a μ-instruction, a decoder is required for each group.

In Figure 13(c): encoding of control signals in m = 2 groups (Group 0 and Group).

Three decoders are needed to generate the control signals.

Degree of parallelism (parallel execution of μ-operations) offered is m = 3.

0.3.1.4 Microinstruction sequencing

There are two options for instruction sequencing

1. Include next microinstruction μI_{i+1} address with in the microinstruction μI_i.
2. Microprogram counter (μPC) to store next executable μ-instruction address.

   We assume μ-instruction sequencing with μPC.

0.3.2 Microinstruction address mapping

Shown in Figure 14.

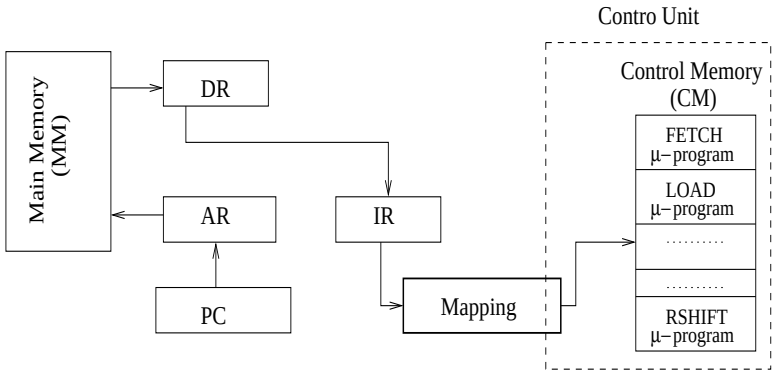


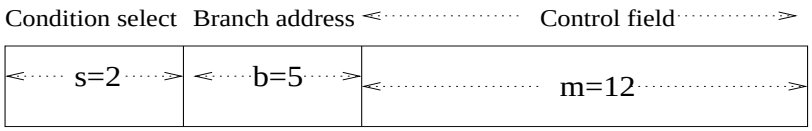
Figure 14: Micro instruction mapping

0.3.3 Microprogrammed control unit

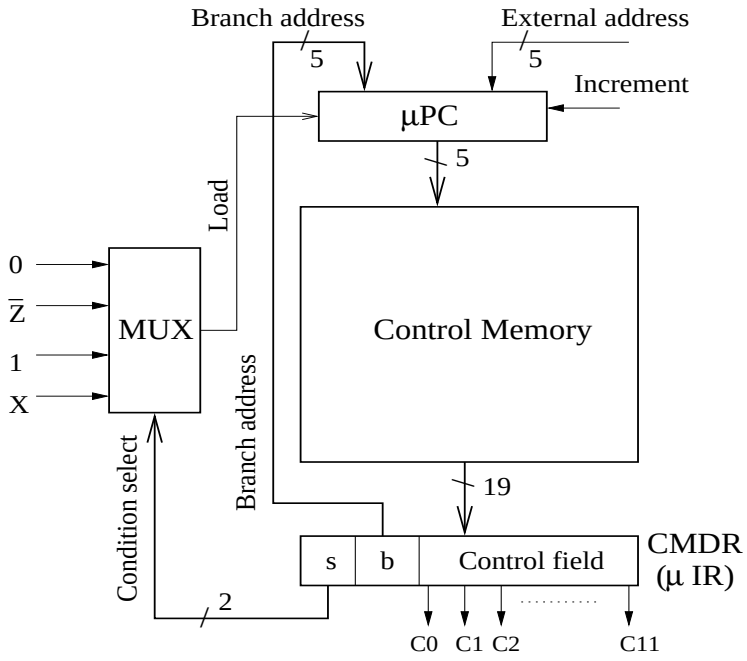
Shown in Figure 15.

Assume horizontal format.

CM size is 32 word. Each of 19 bits. 12-bit is for control field, 5-bit for branch address, and 2-bit for condition select.



a) Micro−instruction format



b) Micro−programmed control unit structure

Figure 15: Micro programmed control unit