Microprogramming

Control unit can be designed using:

- Finite state machine approach.
  - Use state diagrams and transitions between states.
  - Use CAD tools for "hardwiring" control
  - Good if number of states is relatively small
  - Has gotten better throughout the years because of RISC and because better CAD tools
- Microprogramming
  - Use a "programming approach" very much as we decomposed single cycle implementation into multiple (shorter cycles)
  - Good when many states (structured approach), e.g., in CISC machines
  - Implementation with ROM (used to be faster than RAM)

Microprogramming – Basic idea

The (main) data path remains unchanged.

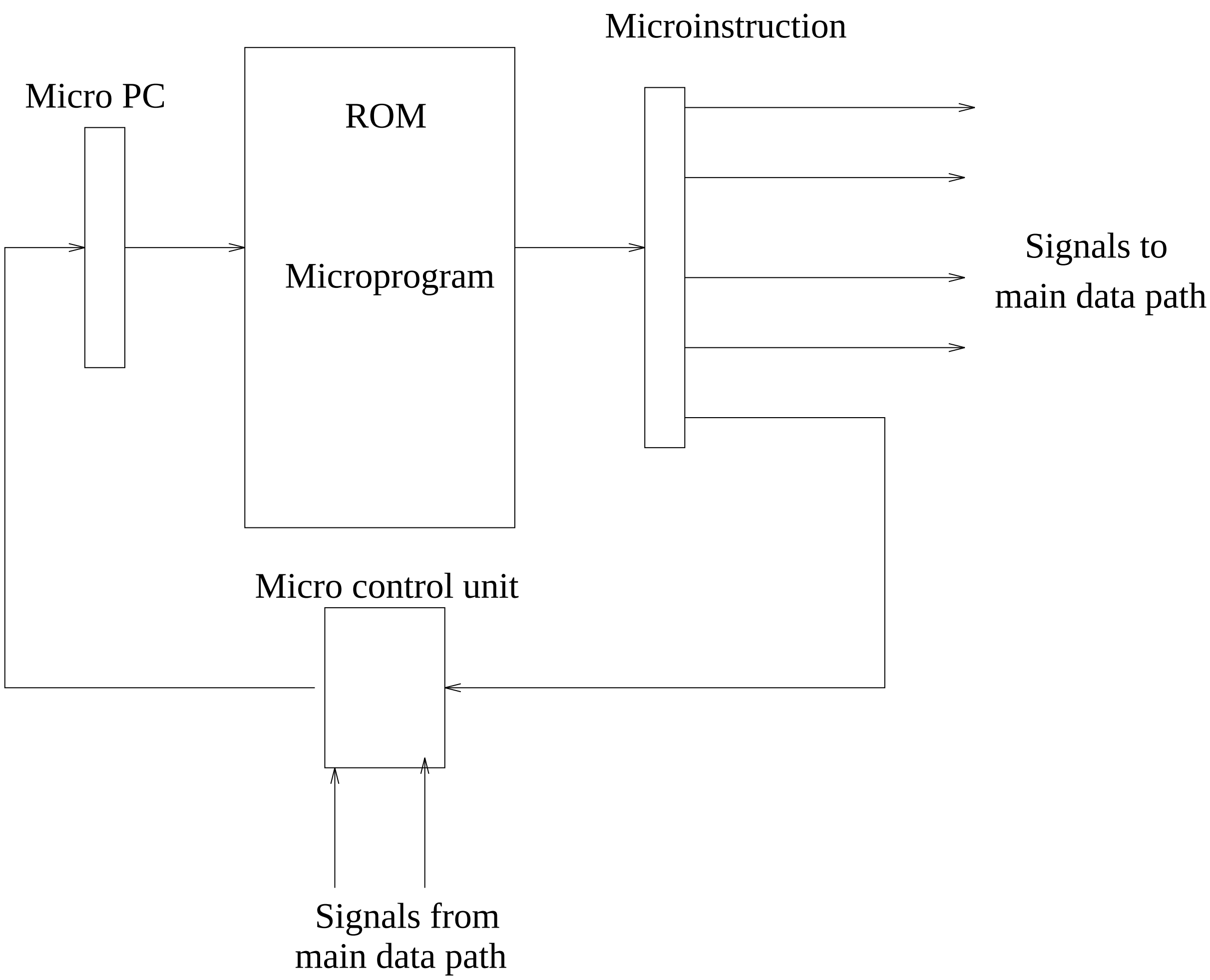
Build a *very simple* auxiliary data path and associated control unit whose role is to generate control signals for the main data path.

The micro data path fetches microinstructions.

The micro control unit sends signals to the main data path.

Each machine language instruction is “interpreted” by a sequence of microinstructions
The interpretation of the ISA is the “microprogram” for the machine.

Microprogramming – Block diagram



Microinstructions

In each microinstruction there is an encoding for what to do about control signals in the main data path, i.e.,

1. ALU control (what is the operation to perform)
2. Select source 1 of ALU
3. Select source 2 of ALU
4. Destination of ALU
5. Memory read or write
6. Memory register (destination for load, source for store)
7. How to modify PC

The microinstruction has also a field to find the next microinstruction

Microinstruction format

Each microinstruction causes one datapath cycle to execute

Each field of the microinstruction contains control bits for one unit in the data path. For example:

ALU	Sr1	Sr2	Dest	mem	mem reg	PCw	sequence

0 means PC is source
1 means rs is source

for finding next microinstruction

0 means 4 is source
1 means get source for extension unit
2 means shift by 2 in adder
3 means rt is source

etc

Microprogramming: Advantages and drawbacks

Advantage – Flexibility

- Can "fix" ISA after data path has been designed
- Can have several ISA's on the same machine
- Can add instructions to an existing ISA

Drawbacks – Cost

- Time penalty for fetching and executing microinstruction
- Cost penalty in chip "real estate"

Microprogramming Implementation

Vertical microcode – Sequential execution

- Highly encoded microinstructions
- Usually the microinstruction has an opcode for one main data path unit (e.g, no concurrent activation of ALU and Memory)
- Looks like ... a RISC instruction set

Horizontal microcode

- Very wide (100+ bits) microinstructions
- A field for each data path unit with little encoding
- Lots of parallelism possible
- Each microinstruction carries the address (modifiable) of its successor