International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Review on the working of the Microprogrammed Control Unit

Ananya Dhulipala[1], K. B. Ramesh[2]

[1]UG Student, Department of Electronics and Instrumentation Engineering, RV College of Engineering, Bengaluru, India. Email: dhulipala.ananya@gmail.com
[2]Associate Professor, Department of Electronics and Instrumentation Engineering, RV College of Engineering, Bengaluru, India. Email: rameshkb@rvce.edu.in

ABSTRACT:

For processing instructions the CPU has a much needed part called the Control Unit. There are two types of Control Unit- Hardwired Control Unit and the Microprogrammed Control Unit.The latter is the most widely used commercially and is highly customisable. Hence, it is very important to understand how it's working. This paper summarises the entire working of the Micro-programmed Control Unit.

Introduction:

The Hardwired Control Unit changes its state to another state at every clock cycle. The output of this unit results in the form of control signals and is determined by logic gates and wires. The Microprogrammed Control Unit works by storing microinstructions in the Control Memory that determines the control signals. It easily handles complex instructions also and the implementation of this control w is less costly. Thus, depending on the aim of the computer the CPU will contain either of these to kinds of Control Units.

The better performing microprogrammed control unit uses dynamic programming. This enables the microprogram to be directly from the auxiliary memory. This modified control unit uses a writable memory for storing microprograms and other related data.Control words represent control variables that are basically a blend of 0s and 1s. Microoperations are low-level instructions used in designs for implementation of complex machine commands. Also termed as micro-instructions. Microinstructions are symbolic microprograms that have been translated to their binary form by an Assembler. Each line of the microprogram that is converted to assembly language represents a microinstruction .

As stated before microprograms are a sequence of microinstructions. Once the Control Unit is in operation no alteration needs to be made to the microprogram. Thus the control memory has to be a Read-Only Memory. The microprogram involves the placement of all control variables in words of the OM that can be further used to operate the control Unit by consecutively reading them which become permanent due to hardware connections made in the unit. Some parts of microinstructions also known as microcodes are accessed multiple times. These can be saved by the use of subroutines.

Organisation of the Microprogrammed Unit:

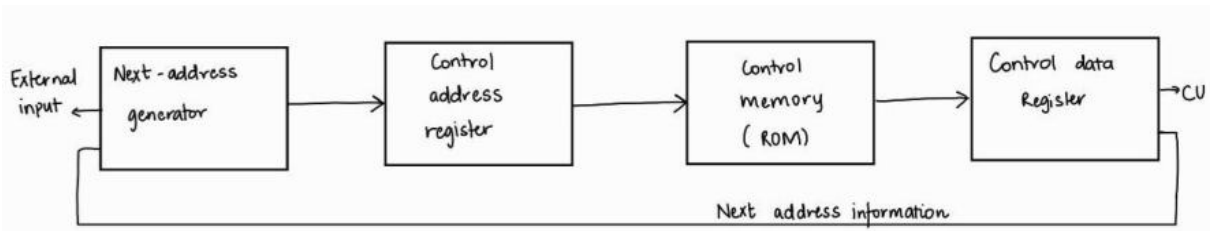


Figure 1: Flowchart of working of the Microprogrammed Control Unit

Figure 1 depicts the general method of working of a Microprogrammed Control Unit. The Control memory is a read-only memory and stores all information permanently.Each address of every microinstruction is read by the control data register and sourced from the control address memory register. The microoperations that are determined from the control words after execution the control will move on to the next address. The next instruction can either be the one that follows the last or can be at any random location.During the execution of the microoperations the following address is computed and then transferred to the Control Address generator. Thus, the process repeats.Hence each microinstruction contains the bits for initiating the operations of the data processor. The next-address generator is called a **microprogram sequencer** whose basic function is to sequentially generate the next address one-by-one from the control address register.

The data register is also known as the pipeline register. It allows the implementation of the microoperations specified by the control words at the same time as the generation of the microinstructions. This requires a 2-phase clock, with one clock applied to the address register and the other one to the data register.

Address Sequencing:

Microinstructions are stored in the form of groups in the control memory. They together form a **routine.** The address sequencing is done in 4 steps:

1. **Incrementing of the control address register:** After the computer is switched on an initial address is loaded into the control address register which is usually in charge of the activation of the fetch routine. At the end of this routine the instructions are present in the instruction register.
2. **Unconditional branch or conditional branch(truth table present later):** The next address is the routine that determines the effective address of the operand. The machine instructions might contain bits that specify the various address modes. After the completion of the effective address computation routine which is present in the control memory the addresses of each operand will be present in the memory address register.
3. **Mapping from bits of instruction to address:** In this step the microoperations that control the execution of the instructions are carried from the memory. The steps followed depend on the operational code that are a part of the microinstructions. The modification from the instruction code to an address present in the control memory that contains the routine is called **mapping.**
4. **Facilitation for subroutine call and return:** After the routine is achieved the microinstructions are sequenced. The microprogram uses ROM as an external memory for storing the return addresses. after the execution control returns to the fetch routine by execution of an unconditional branch to the first address of the fetch routine.

Control Memory: Address Selection

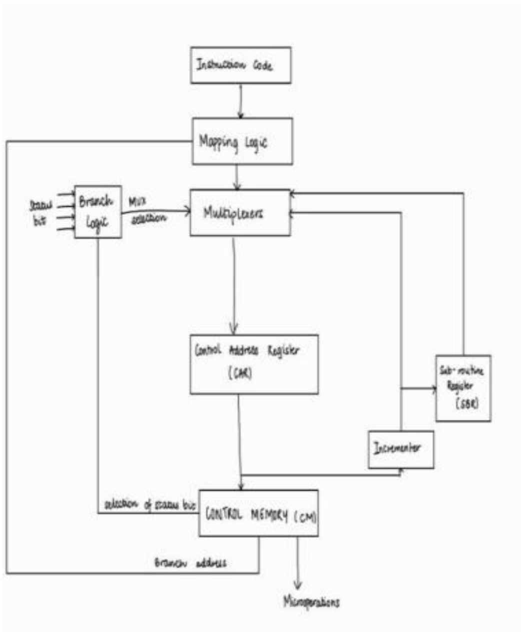
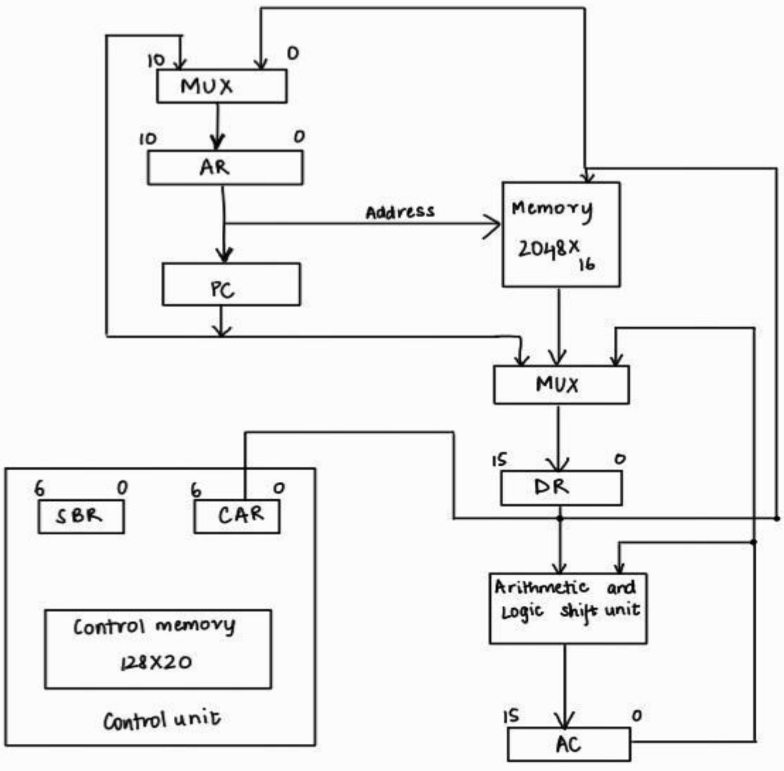


Figure 2: Selection of Address for Computer Register

The block diagram as shown in Figure 2 depicts the control memory and associated hardware required for selection of the addresses of the microinstructions.These instructions contain the signals to execute the operations and also to generate the next address. As shown in the block diagram, The control address register (CAR) receives the address by 4 different paths.

The incrementer increments the sequence so that all the microinstructions can be traversed. Branching is done by specifying the branch address in one of the fields of the instruction. Conditional branching is done by selection of a part of the status bit using a microinstruction and hence determining it's condition.A mapping logic circuit helps in transferring the external address. The subroutine return address value serves the purpose of as and and when the computer wants to return from the subroutine. The branch logic in the above given figure provides decision-making capabilities.



Configuration of Hardware present inside the Control Unit

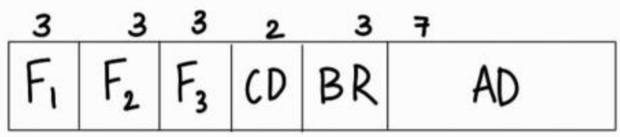
Figure 3: Flowchart regarding the hardware inside the Computer

The above block diagram depicts the entire hardware present inside the CPU.As depicted in the above flowchart, 2 memory units are present out of which one of them stores the data and the other is to store the microprogram. It also contains 6 registers - 4 Processor Unit Registers and 2 Control Unit Registers. The processor unit registers are the Accumulator , Program Counter , Address Register and the Data Register .Whereas the Control unit Registers are the Control Address Register and the Subroutine Register.

The multiplexers are used as a replacement to the common bus. They are mainly used for transferring data between the registers.

Microinstruction Format

The first 3 named F1, F2 and F3 as depicted in the diagram. These are responsible for the microoperations for the computer. These are further divided



into 3 bits each that encode seven distinct microoperations which gives us a total of 21 microoperations.

The CD field has 2 bits that specify 4 status bit conditions. The below table shows information about the status conditions.

Table 4.1: Condition Field

CD	Symbol	Condition	Condition
00	U	Always = 1	Unconditional branch
01	I	DR	Indirect Address bit
10	S	AC	Sign bit of AC
11	Z	AC	Zero value in AC

The BR field specifies the type of branch to be used. The table 4.2 specifies which branches according to the branch code.

Table 4.2: Branch Field

BR	Symbol	Function
00	JMP	AD points to CAR if CD==1
		Else CAR+1 points to CAR if CD==0
01	CALL	AD points to CAR and CAR+1 points to SBR if CD==1
		Else only CAR+1 points to CAR if CD==0
10	RET	return from subroutine
11	MAP	The address DR from 11-14 points to CAR from 2-5 and the rest are 0

Microprogram Sequencer

Other than the control memory the microprogrammed control unit has another essential part called the microprogram sequencer. It's main function is to select the net address. It can be customised with an internal organisation with multiple digital functions to suit a wide range of purposes so that it can be broadly accepted. Most commercial control sequencers also have a register stack present inside that is used for storing addresses temporarily.

Conclusion

In this paper we have presented an overview of the organisation of the Microprogrammed Control Unit and it's working. The choice of particular instructions can be made by customising it as per needs. Thus owing to the flexible nature of this type of control unit. But as studied in the above paper, the time it takes to fetch microinstructions from the control memory is considerably high and thus the micro-programmed control unit is slower in speed as fetching of the microinstructions takes longer.

References

1. COMPUTER SYSTEM ARCHITECTURE, MORRIS M. MANO, 3RD EDITION, PRENTICE HALL INDIA.
2. By Carl Hamacher Zvonko Vranesic, Safwat Zaky - International Edition of Computer Organisation Fifth Edition.
3. Thesis on A Segmented Microprocessor by Steven Vaillan Court. https://ruor.uottawa.ca/bitstream/10393/3834/1/MK48647.PDF