Computer Architecture and Organization Mini Project CS - 205

CONTROL UNIT

Design and Analysis of Hardwired Control Unit

Group 12

Rahul Shah : 1815057

Pocham Umesh Chandra : 1815058

Swagat S. Bhuyan : 1815059

Anamitra Saikia : 1815060

4th Semester CSE

Under Supervision

of

Dr. Malaya Dutta Borah

Department of Computer Science

and

Engineering

TABLE OF CONTENTS

GLOSSARY

Finite State Machine

A Finite State Machine (FSM) or Finite Automaton (FA) is a simple idealized machine used to recognize patterns within input taken from some character set (or alphabet) Σ. It is an abstract machine that can be in exactly one of a finite number of states at any given time. The FSM can change from one state to another in response to some inputs; the change from one state to another is called a transition.

Karnaugh Map

The K-map method of solving the logical expressions is referred to as the graphical technique of simplifying Boolean expressions. The Karnaugh map reduces the need for extensive calculations by taking advantage of humans' pattern-recognition capability.

Control Word

Any given datapath will have a number of control signals. By asserting or de-asserting these control signals at different times, the datapath can perform different register-transfer operations. Since the execution of an operation requires the correct assertion and deassertion of all of the control signals together they are considered as a unit rather than as individual signals. All of the control signals for a datapath, when grouped together, are referred to as a control word. Hence, a control word will have one bit for each control signal in the datapath.

D Flip Flop

The D Flip Flop or Data FF is a Flip Flop that simply passes on the 1 bit data (either 0 or 1) onto the ongoing circuit. The D type flip-flop has one data input 'D' and a clock input. The circuit edge triggers on the clock input. The flip-flop also has two outputs Q and Q' (where Q' is the reverse of Q).

ABSTRACT

Control Unit is the part of the computer's central processing unit (CPU), which directs the operation of the processor. It fetches internal instructions of the programs from the main memory to the processor instruction register, and based on this register contents, the control unit generates a control signal that supervises the execution of these instructions. A control unit works by receiving input information to which it converts into control signals, which are then sent to the central processor. The computer's processor then tells the attached hardware what operations to perform.

Hardwired Control Unit focuses on the practical circuit approach to designing a fully functioning Control Unit. This includes decoders, logic gates, etc. facilitating a more hands-on approach. In this Project, we sort to find a suitable State Machine to replicate the workings of an optimized Hardwired Control Unit. For this, we ball park a rough idea for the State Diagram following an algorithmic flowchart to make the final set of states and transitions. We then Proceed to Solve the State Tables for the various Variable Flip Flop States in order to conjure the right circuitry required to simulate the required Hardwired Control Unit (For this, we use Karnaugh Maps)

The Control Unit is considered to be the 'brain' of the CPU, as it manages all the functionalities and time manipulations of various tasks that a CPU handles. Serving such an important purpose for the CPU, the Control Unit and its quality plays a big role when deciding the status of a given CPU.

INTRODUCTION

Computers have come a long way from mechanical devices capable of maybe one calculation per second, to CPUs running at kilohertz and megahertz speeds. In the early days of electronic computing, processors were typically made faster by improving the switching time of the transistors inside the chip - the ones that make up all the logic gates, ALUs, Control Units, etc. But just making transistors faster and more efficient only went so far, so processor designers have developed various techniques to boost performance allowing not only simple instructions to run fast, but also performing much more sophisticated operations using circuits instead of using simple ALU instructions. This is what is covered in making an advanced CPU – the architecture and organization of various aspects that make up a CPU.

A central processing unit (CPU) is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions. The Design of a CPU can be basically broken down to the following components:

- ALU (Arithmetic and Logical Unit)
- Registers
- Memory and I/O interfaces
- Control Unit
- Clock
- Buses

Control Unit is the part of the computer’s central processing unit (CPU), which directs the operation of the processor. It was included as part of the Von Neumann Architecture by John von Neumann. It is the responsibility of the Control Unit to tell the computer’s memory, arithmetic/logic unit and input and output devices how to respond to the instructions that have been sent to the processor.

It fetches internal instructions of the programs from the main memory to the processor instruction register, and based on this register contents, the control unit generates a control signal that supervises the execution of these instructions.

A control unit works by receiving input information to which it converts into control signals, which are then sent to the central processor. The computer's processor then tells the attached hardware what operations to perform. The functions that a control unit performs are dependent on the type of CPU because the architecture of CPU varies from manufacturer to manufacturer.

- It coordinates the sequence of data movements into, out of, and between a processor's many sub-units.
- It interprets instructions.
- It controls data flow inside the processor.
- It receives external instructions or commands to which it converts to a sequence of control signals.
- It controls many execution units (i.e. ALU, data buffers and registers)
- It also handles multiple tasks, such as fetching, decoding, execution handling and storing results.

HISTORY OF PROCESSOR DEVELOPMENT

The foundation of processors occured after the groundbreaking discovery of semiconductor materials. Silicon was discovered by **Baron Jons Jackob Berzelius** in 1823 i.e one year after the first mechanical computer made by Charles Babbage. This proved to be essential as Silicon had enormous potential in this field, while simultaneously being abundant in nature.

In 1903, Nikola Tesla patented the Logic Gates and Switches, marking a new chapter in the world of computing. This Groundbreaking invention facilitated the potential for complex machines capable of heavy calculations simply not limited to the human brain.

A new wave of innovations came up around the 1940s which made possible the categorizing of the rich history of computers into five definable generations:

GENERATIONS of COMPUTER

First Generation - Vacuum Tubes (1940 - 1956)

These early computers used vacuum tubes as circuitry and magnetic drums for memory. As a result they were enormous, literally taking up entire rooms and costing a fortune to run. These were inefficient materials which generated a lot of heat, sucked huge electricity and subsequently generated a lot of heat which caused ongoing breakdowns.

These first generation computers relied on 'machine language' (which is the most basic programming language that can be understood by computers). These computers were limited to solving one problem at a time. Input was based on punched cards and paper tape. Output came out on print-outs. The two notable machines of this era were the UNIVAC and ENIAC machines – the UNIVAC is the first ever commercial computer which was purchased in 1951 by a business – the US Census Bureau.

Second Generation – Transistors (1956 – 1963)

The replacement of vacuum tubes by transistors saw the advent of the second generation of computing. Although first invented in 1947, transistors weren't used significantly in computers until the end of the 1950s. They were a big improvement over the vacuum tube, despite still subjecting computers to damaging levels of heat. However they were hugely superior to the vacuum tubes, making computers smaller, faster, cheaper and less heavy on electricity use. They still relied on punched cards for input/printouts.

The language evolved from cryptic binary language to symbolic ('assembly') languages. This meant programmers could create instructions in words. About the same time high level programming languages were being developed (early versions of COBOL and FORTRAN). Transistor-driven machines were the first computers to store instructions into their memories – moving from magnetic drum to magnetic core 'technology'. The early versions of these machines were developed for the atomic energy industry.

Third Generation – Integrated Circuits (1964 – 1971)

By this phase, transistors were now being miniaturised and put on silicon chips (called semiconductors). This led to a massive increase in speed and efficiency of these machines. These were the first computers where users interacted using keyboards and monitors which interfaced with an operating system, a significant leap up from the punch cards and printouts. This enabled these machines to run several applications at once using a central program which functioned to monitor memory.

As a result of these advances which again made machines cheaper and smaller, a new mass market of users emerged during the '60s.

Fourth Generation – Microprocessors (1972 – 2010)

This was the era of Very Large Scale Integration (VLSI), the 4th generation. This revolution can be summed in one word: Intel. The chip-maker developed the Intel 4004 chip in 1971, which positioned all computer components (CPU, memory, input/output controls) onto a single chip. What filled a room in the 1940s now fit in the palm of the hand. The Intel chip housed thousands of integrated circuits. The year 1981 saw the first ever computer (IBM) specifically designed for home use and 1984 saw the MacIntosh introduced by Apple. Microprocessors even moved beyond the realm of computers and into an increasing number of everyday products.

The increased power of these small computers meant they could be linked, creating networks. Which ultimately led to the development, birth and rapid evolution of the Internet. Other major advances during this period have been the Graphical user interface (GUI), the mouse and more recently the astounding advances in lap-top capability and hand-held devices.

Fifth Generation – Artificial Intelligence (2010 -)

Computer devices with artificial intelligence are still in development, but some of these technologies are beginning to emerge and be used such as voice recognition. AI is a reality made possible by using parallel processing and superconductors. Leaning to the future, computers will be radically transformed again by quantum computation, molecular and nanotechnology. The essence of the fifth generation will be using these technologies to ultimately create machines which can process and respond to natural language, and have capability to learn and organise themselves.

HISTORY OF MICROPROCESSORS

The microprocessor is a single IC package in which several useful functions are integrated and fabricated on a single silicon semiconductor chip. Its architecture consists of a central processing unit, memory modules, a system bus, and an input/output unit.

Just like the CPU, the history of microprocessor can also be categorized into generations marking evolution in a step by step manner

1st Generation

This was the period from 1971 to 1973 of microprocessor's history. In 1971, INTEL created the first microprocessor 4004 that would run at a clock speed of 740 kHz. During this period, the other microprocessors in the market including Rockwell international PPS-4, INTEL-8008, and National semiconductors IMP-16 were in use. But, all these were not TTL compatible processors.

2nd Generation

This was the period from 1973 to 1978 in which very efficient 8-bit microprocessors were implemented like Motorola 6800 and 6801, INTEL-8085 and Zilog-Z80, which were among the most popular ones. Owing to their super fast speed, they were costly as they were based on NMOS technology fabrication.

3rd Generation

During this period 16-bit processors were created and designed using HMOS technology. From 1979 to 1980, INTEL 8086/80186/80286 and Motorola 68000 and 68010 were developed. Speeds of those processors were four times better than the 2nd generation processors.

4th Generation

From 1981 to 1995 this generation developed 32-bit microprocessors by using HCMOS fabrication. INTEL-80386 and Motorola's 68020/68030 were the popular processors.

5th Generation

From 1995 to until now this generation has been bringing out high-performance and high-speed processors that make use of 64-bit processors. Such processors include Pentium, Celeron, Dual and Quad-core processors.

5th GEN INTEL FAMILIES

INTEL CELERON

Intel Celeron was introduced in April 1998. It refers to a range of Intel's X86 CPUs for value personal computers. It is based on Pentium 2 and can run on all IA-32 computer programs.

INTEL PENTIUM

Pentium was introduced on March 2, in 1993. Pentium succeeded the Intel 486; The 4 indicates the fourth generation microarchitecture in the microprocessor's history. Pentium refers to Intel's single-core x 86 microprocessor, which is based on the fifth-generation micro-architecture. This processor's name was derived from the Greek word Penta, meaning five.

The original Pentium processor was succeeded by the Pentium MMX in 1996. This processor has a data bus of 64 bits. A standard single transfer cycle can read or write up to 64 bits at a time. The Burst read and writes back cycles are supported by the Pentium processors. These cycles are used for cache operations and transfer 32 bytes (size of the Pentium cache line) in 4 clocks. All cache operations are burst cycles for the Pentium.

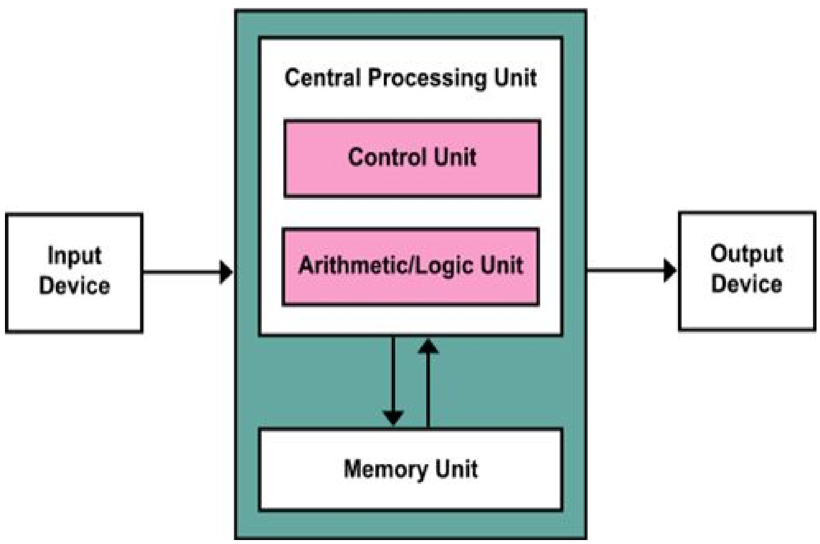
XEON

Xeon processor is a 400 MHz Pentium processor from Intel for use in workstations and enterprise servers. This processor is designed for multimedia applications, engineering graphics, Internet and large database servers.

THE CENTRAL PROCESSING UNIT

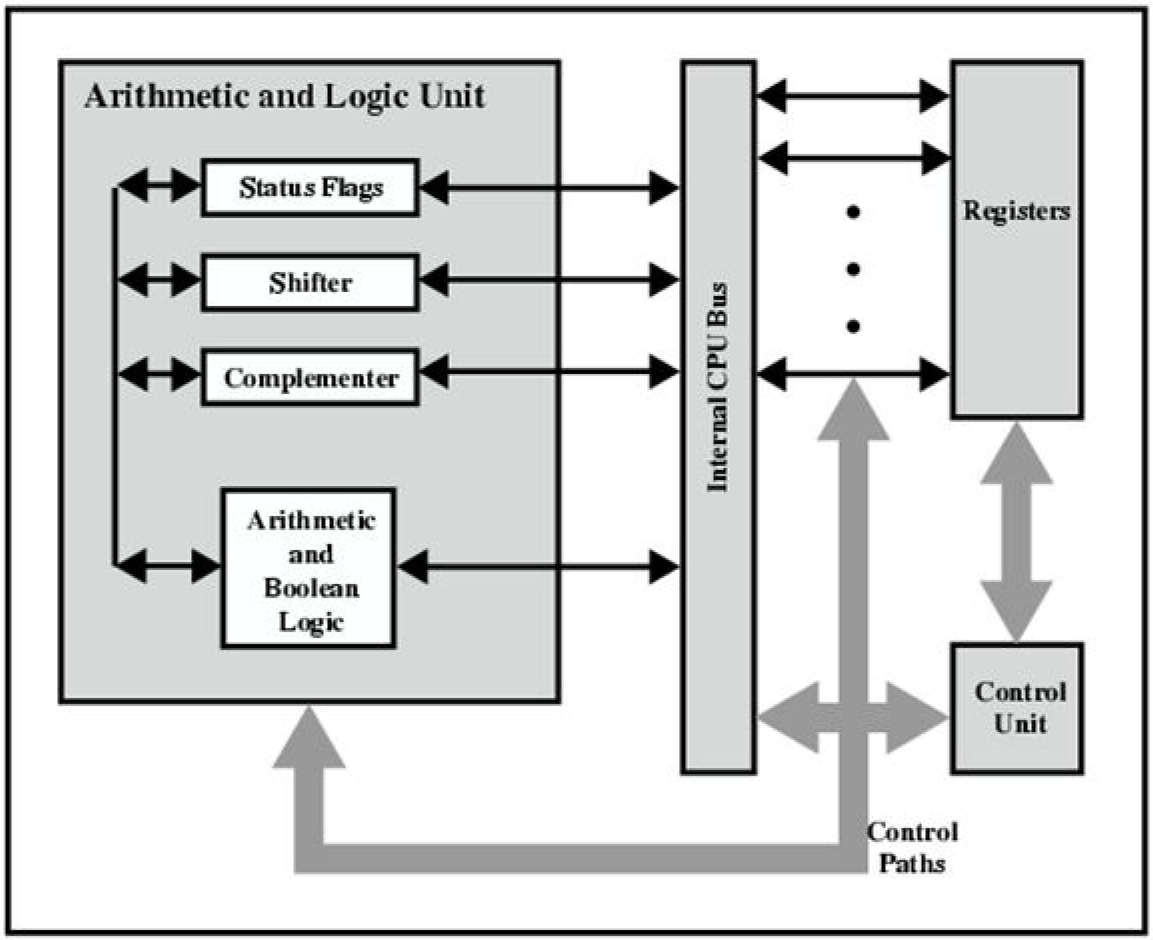
INTRODUCTION

The **central processing unit (CPU)** of a computer is a piece of hardware that carries out the instructions of a computer program. It performs the basic arithmetical, logical, and input/output operations of a computer system. It is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions.



As discussed earlier, the aspects that make up a CPU can be basically broken down to the following components:

- ALU (Arithmetic and Logical Unit)
- Registers
- Memory and I/O interfaces
- Control Unit
- Clock
- Buses



The Diagram above represents the internal organization of a CPU.

CPU CORE BASED CATEGORIZATION

Single Core CPU

The oldest type of computer CPU available are single-core CPUs, and this was initially the only CPU type that could be used on computers.
Single-core CPUs can only start one task at a time, so multitasking wasn't very successful. This means output declines were noticeable every time more than one application was running. Since only one operation could be started at a time, another could be triggered before the first one was done but the machine would run more slowly with every new operation.

Dual Core CPU

A dual-core Processor has two main CPUs and therefore operates like two single CPUs. In comparison, if more than one operation is performed, the Dual-core CPUs can perform several tasks more effectively, whereas in the single Core CPU the processor has to move between the various data-stream sets. To optimize the use of a dual-core CPU, a specialized code, called SMT (simultaneous multi-threading technology), needs to be implemented on the operating system and the programs working on it. Dual-core CPUs are faster than core processors, but not very fast as quad-core CPUs.

Quad Core CPU

Quad-core CPUs are multi-core CPUs with four cores on a single CPU. Like two core CPUs, quad cores will break the workload between four cores, so much more work is done with the quad. Every core is connected to other circuits inside the chip, such as cache, memory and I / O port management. Such kinds of CPUs are beneficial for people who need to run several different programs simultaneously and for gamers.

ARITHMETIC AND LOGICAL UNIT (ALU)

The ALU is the mathematical brain of a computer. It is a combinational digital electronic circuit that performs arithmetic and bitwise operations on integer binary numbers. It is a fundamental building block of many types of computing circuits, including the central processing unit (CPU) of computers, FPUs, and graphics processing units (GPUs). A single CPU, FPU or GPU may contain multiple ALUs.

ALUs in its early stages had fewer instructions, hence fewer unique circuits. This meant that to perform complex calculations, the CPU relied on multiple cycles of basic ALU instruction circuits to get the job done. However with higher generations, calculational complexities sky-rocketed, hence the need for an ALU which could handle bigger calculations without compromising on clock cycles. This was achieved by implementing different Adder and Multiplier circuits along with the basic logic gates into the ALU of modern CPUs.

An ALU, like it sounds, consists of the Logical unit and the Arithmetic unit.

LOGICAL UNIT

This section of the ALU consists of all the logic gates which deal with logical operations, which although carry out bit manipulations, also help out in basic calculations as well, thanks to the versatility in logic gates when designed into a circuit. The basic Logic gates are:

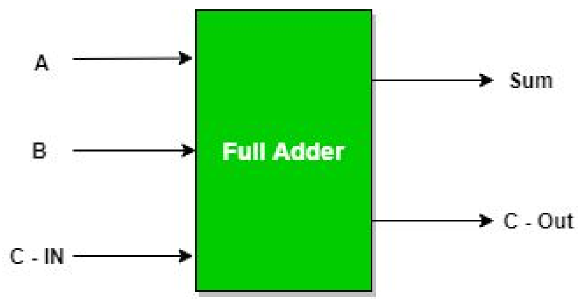
- AND gate
- OR gate
- NOT gate
- NAND gate
- NOR gate
- XOR gate
- XNOR gate

ARITHMETIC UNIT

Deals with the arithmetic aspect of CPU calculations, using simple adder circuits with simpler cycles, and complex multipliers to reduce clock cycles for instructions.

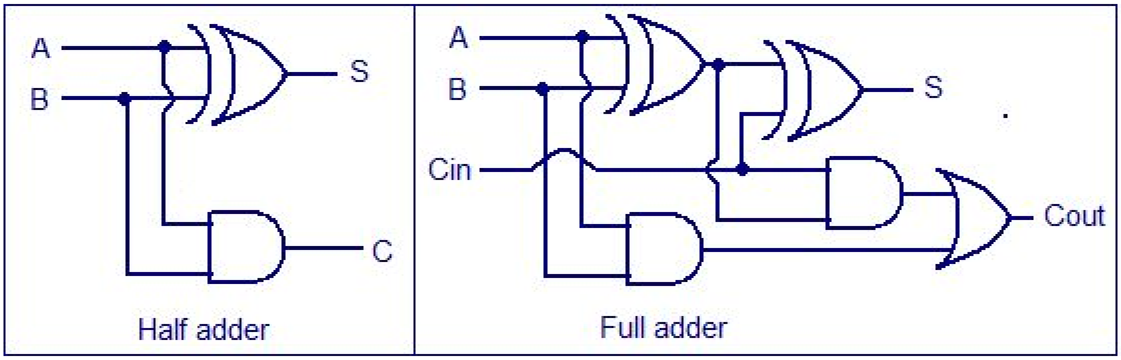
- HALF ADDER:
    - A half adder just adds two bits together and gives a two-bit output.
- FULL ADDER:
    - Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.

A full adder logic is designed in such a manner that it can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to the other.



Inputs			Outputs	
A	B	C – IN	Sum	C - Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

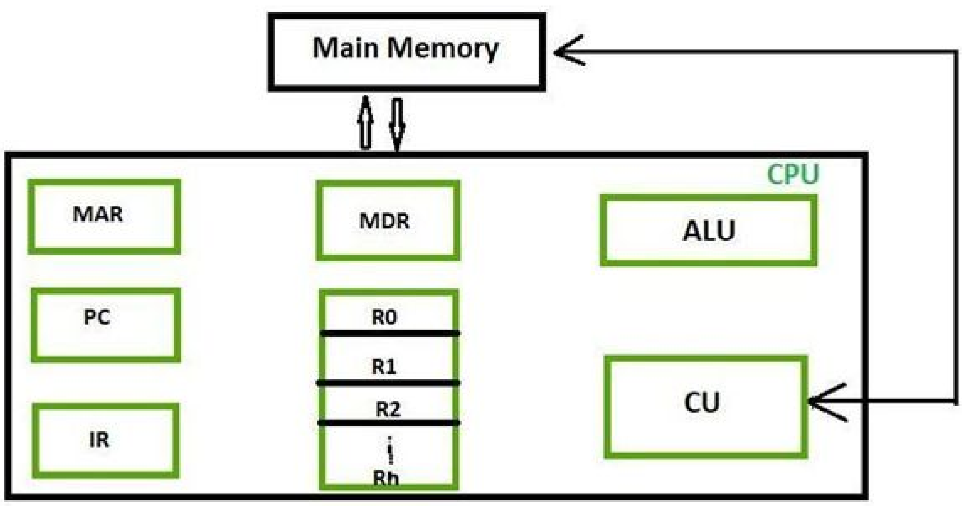
A Full adder can be made by combining two half adder circuits together (a half adder is a circuit that adds two input bits and outputs a sum bit and a carry bit).



REGISTER ORGANIZATION

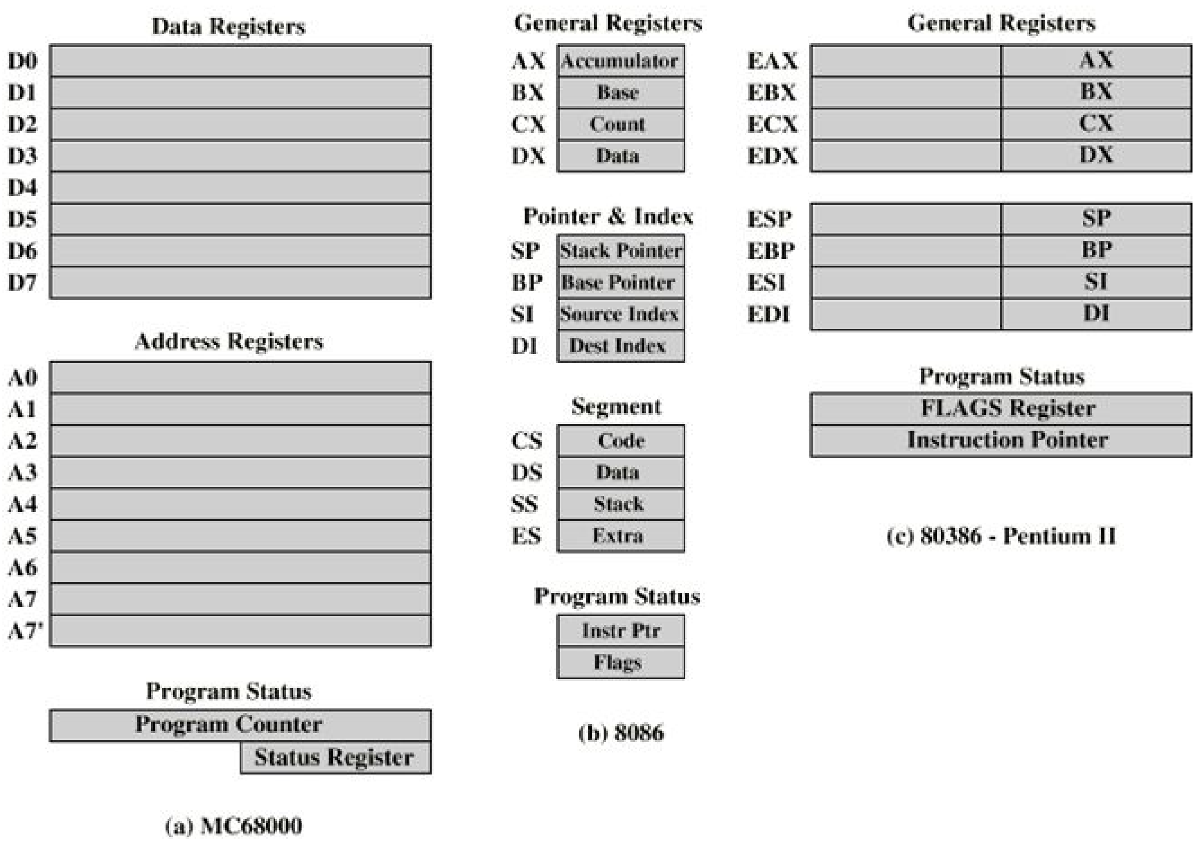
INTRODUCTION

In Computer Architecture, the Registers are very fast computer memory which are used to execute programs and operations efficiently. Within the CPU, registers function as a level of memory above main memory and cache in the hierarchy. This is done by giving access to commonly used values, i.e the values which are at the point of operation/execution at that time.



WORKING

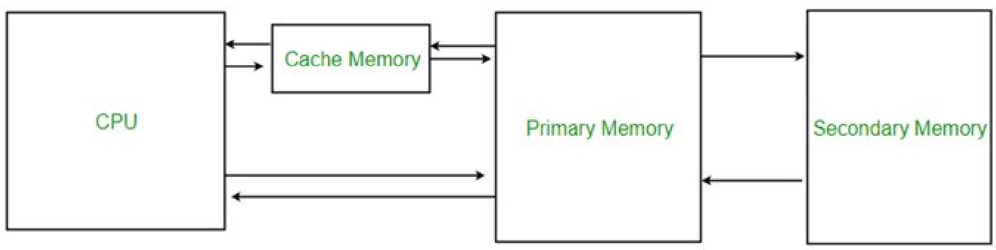
Typically, the CPU updates the PC after each instruction fetch so that the PC always points to the next instruction to be executed. A branch or skip instruction will also modify the contents of the PC. The fetched instruction is loaded into an IR, where the opcode and operand specifiers are analyzed. Data is exchanged with memory using the MAR and MBR. In a bus-organized system, the MAR connects directly to the address bus, and the MBR connects directly to the data bus. User-visible registers, in turn, exchange data with the MBR.



CACHE MEMORY ORGANIZATION

Cache Memory is a special very high-speed memory. It is used to speed up and synchronize with a high-speed CPU. Cache memory is costlier than main memory or disk memory but economical than CPU registers. Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU. It holds frequently requested data and instructions so that they are immediately available to the CPU when needed.

Cache memory is used to reduce the average time to access data from the Main memory. The cache is a smaller and faster memory which stores copies of the data from frequently used main memory locations. There are various different independent caches in a CPU, which store instructions and data.



INPUT OUTPUT ARCHITECTURE

The I/O subsystem of a computer provides an efficient mode of communication between the central system and the outside environment. It handles all the input-output operations of the computer system.

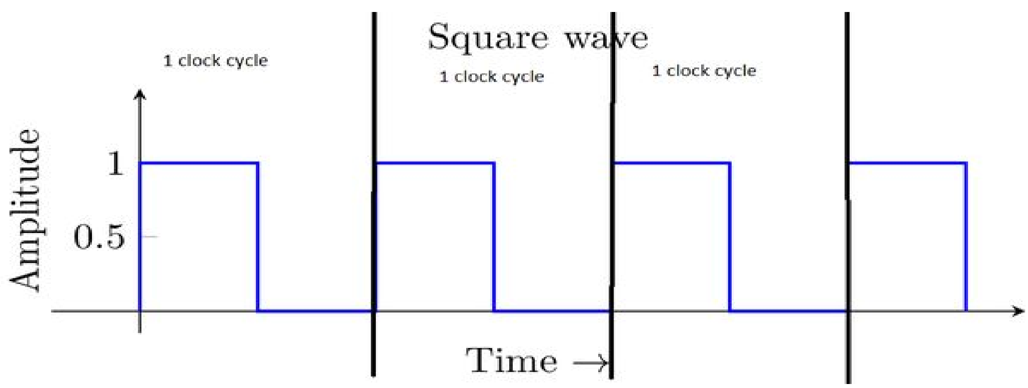
CONTROL UNIT

Control Unit is the part of the computer's central processing unit (CPU), which directs the operation of the processor. It was included as part of the Von Neumann Architecture by John von Neumann. Being the crux of this mini project, it shall be explained in detail in the following sections as it also has a crucial role to play ultimately in the technicalities of this project, since we are after all designing a circuit to display the behavior of the Control Unit.

CLOCK

The clock is the timing device that regulates the movements of data throughout the machine, making it a crucial part in designing a CPU. The cycle is a complete action of the clock. It produces a signal by switching on a voltage and then switching it off again, very quickly. One switch on and one switch off is a complete cycle.

Virtually every CPU ever designed is driven by what is called a clock. Basically, a clock is a small circuit that generates a square wave signal: i.e. it generates a fixed length pulse at fixed intervals.

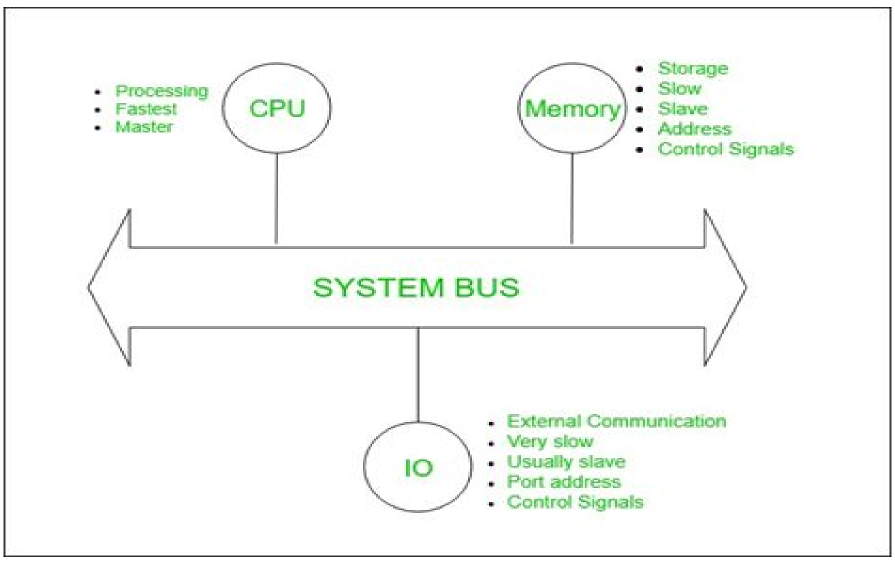


Getting the Clock right is key to determining the speed at which instructions are processed and overall speed of the CPU.

SYSTEM BUS

A bus is a subsystem that is used to connect computer components and transfer data between them. A bus may be parallel or serial. Parallel buses transmit data across multiple wires. Serial buses transmit data in bit-serial format.

Bus is a communication channel. The Characteristics of a bus is shared transmission media. Limitation of a bus is only one transmission at a time. A bus which is used to provide the communication between the major components of a computer is called a System bus.



CONTROL UNIT

INTRODUCTION

The control unit (CU) is a component of a computer's central processing unit (CPU) that directs the operation of the processor during the execution of a program. The control unit fetches internal instructions of programs from the main memory to the processor (computer) instruction register and, based on this register contents, generates control signals that supervise execution of these instructions. The control signals are distributed to all smaller and larger elements of the computer that participate in execution of instructions and need to be controlled. The control signals are usually transmitted by the part of the overall system bus called the control bus.

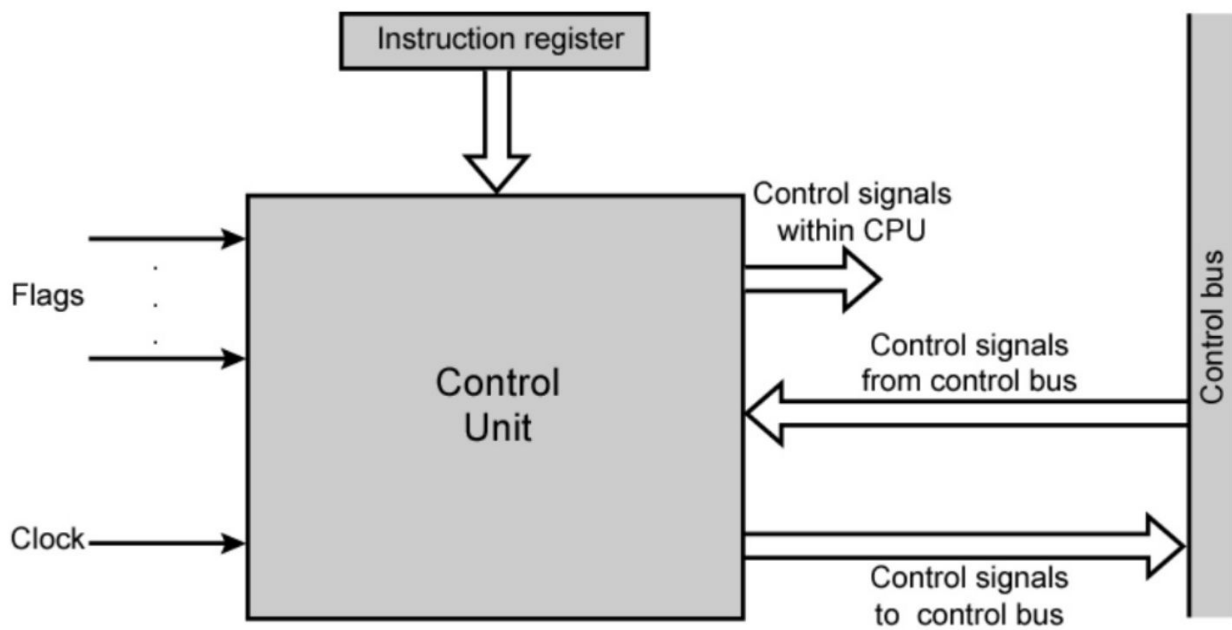


Fig : Model of a Control Unit

Control Unit consists of instruction registers, control signals within the CPU, control signals to/from the bus, control bus, input flags, and clock signals. The information of the control unit or data is temporarily or permanently stored in the control memory. Control memory is of two types. They are Random Access Memory(RAM) and Read-Only Memory(ROM).

Functions of A Control Unit

1. It directs the flow of data sequence between the Central processing Unit and other devices
2. It interprets the instructions and controls the flow of data in the processor.
3. It generates the sequence of control signals from the received instructions from the instruction register.
4. It is responsible for control of execution units such as ALU , Data buffers and Register in the CPU
5. It has the ability to fetch, decode, handle the execution, and store results
6. It communicates with the input and output devices and controls all the units of the computer.
7. Control Unit Does not process and store data

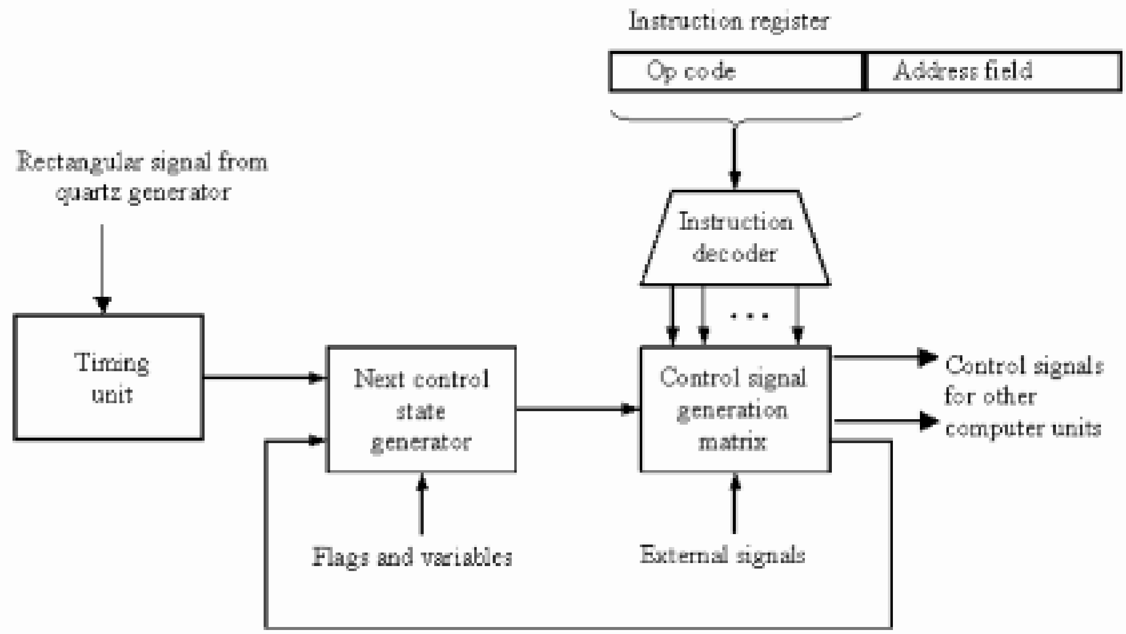
Types Of Control Unit

There are two types of control units in computers:

- Hardwired Control Units
- Microprogrammable (microprogrammed) Control Units.

Hardwired Control units

In Hardwired control units, the control signals are generated by a special hardware logic circuit without any change in the structure of the circuit in which the signal generation method can not be modified without physical change (redesign) of the circuit structure. It is implemented with the help of gates, flip flops, decoders etc. in the hardware. The inputs to the control unit are the instruction register, flags, timing signals etc.



Block diagram of a hardwired control unit of a computer

Functioning of Hardwired Control units

The Data for control signal generation are contained in the operation code (op code) of an instruction.The operation code is decoded in the instruction decoder. The Instruction decoder consists of a set of decoders that decodes different fields of the instruction opcode. As a result, it gives a sequenced activation of output lines which are connected as inputs of the matrix that generates control signals for executive units of the computer.

This matrix implements logical combinations of the decoded signals from the instruction opcode with the outputs from the matrix that generates signals representing consecutive control unit states and with signals coming from the outside of the processor, e.g. interrupt signals.

Control signals for an instruction execution have to be generated during the entire time interval that corresponds to the instruction execution cycle. The appropriate sequence of internal states is organized in the control unit. A number of signals generated by the control signal generator matrix is fed back to inputs of the next control state generator matrix. This matrix combines these signals with the timing signals generated by the timing unit based on the rectangular patterns usually supplied by the quartz generator.

When a new instruction arrives to the control unit, the control unit is in the initial state of new instruction fetching. Instruction decoding results in the control unit to enter the first state according to the execution of the new instruction, which lasts as long as the timing signals and other input signals such as flags and state information of the computer remain unchanged. A change of any of the mentioned signals stimulates the change of the control unit state. This registers as a new input is generated for the control signal generator matrix.

When an external signal appears, e.g. an interrupt, the control unit enters a predetermined control state that is the state concerned with the reaction to this external signal i.e. interrupt processing.The values of flags and state variables of the computer are used to select appropriate states for the instruction execution cycle. The last states in the cycle are control states that initiate fetching the next instruction of the program. When the current instruction is the stop instruction that ends program execution, the control unit enters an operating system state, in which it waits for a next user directive.

For example Microprocessors of the RISC type, designed by DECAlpha, Hewlett-Packard, Compaq, SUN companies, have hardwired control units.

Advantages of Hardwired control unit

- RISC processors are known for their speed, Reduced instruction set and support for limited addressing modes. Hardwired control units are now widely employed in RISC processors.
- They operate at very high speed.
- They are more power efficient than the counterparts.
- Faster than the microprogrammed control unit.
- It simplifies the design of the control unit.

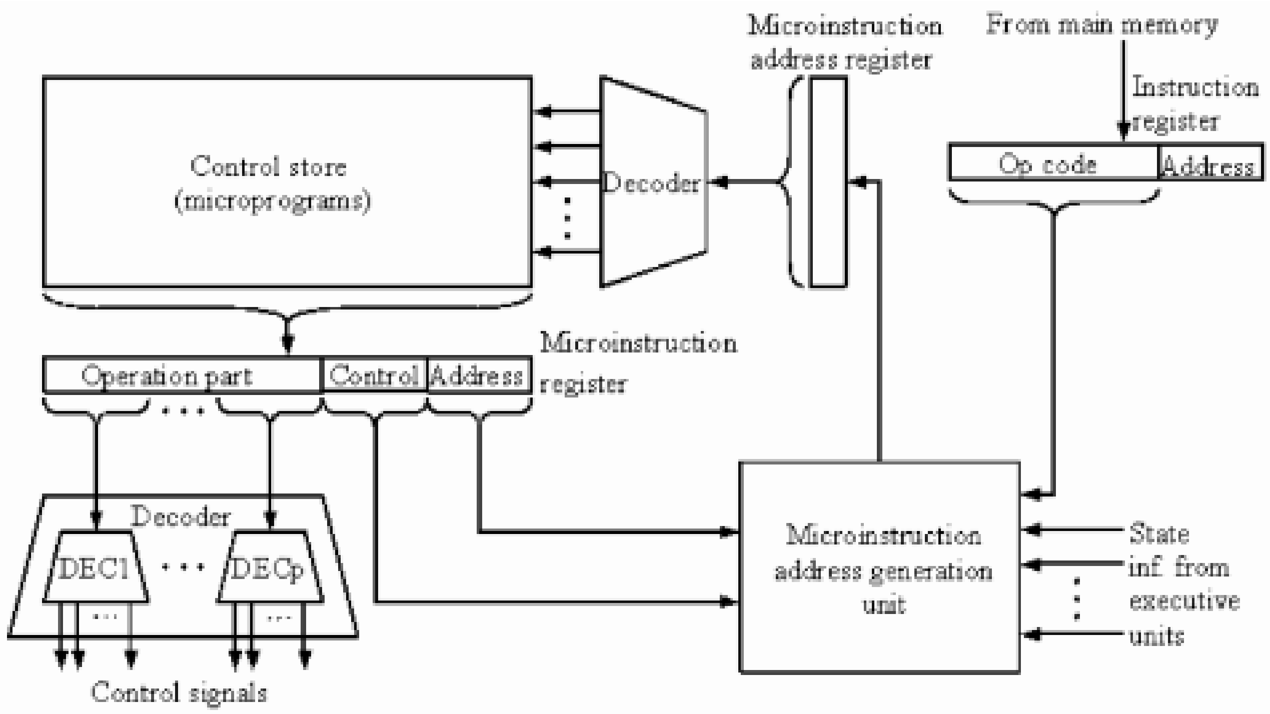
Disadvantages of Hardwired control unit

- It has complex sequencing and micro operation logic.
- It is difficult to design and test.
- It is Difficult to add new instructions.
- If the design has to be changed or modified then it requires change in wiring.
- The occurrence of an error is more.

Microprogrammable Control Units

The Microprogrammed Control unit is implemented by using the programming approach. In Microprogrammed Control, the micro-operations are performed by executing a program consisting of micro-instructions. The Microprogrammed Control Unit stores words containing encoded control signals necessary for instruction execution. In microprogrammed control units, subsequent instruction words are fetched into the instruction register in a usual way. However, the operation code of each instruction is not directly decoded to enable immediate control signal generation but it constitutes the initial address of a microprogram contained in the control store.

Functioning of Microprogrammable Control Units



Microprogrammed control unit with a single level control store

The instruction operation code (op code) from the instruction register is sent to the control store address register. According to this address, the first microinstruction of a microprogram that interprets execution of the required instruction is read to the microinstruction register. This microinstruction contains the encoded control signals in several bit fields. The fields are decoded in a set microinstruction field decoder. Besides the encode control signal fields, the microinstruction contains the address of the next microinstruction of the given instruction microprogram and a control field used to control activities of the microinstruction address generator. The last mentioned field determines the addressing mode (addressing operation) to be applied to the address embedded in the current microinstruction. In microinstructions with the conditional addressing mode, this address is modified with the use of the processor condition flags that represent the status of computations in the current program. The last microinstruction in the microprogram of a given instruction is the microinstruction that fetches the next instruction from the main memory to the instruction register.

In Control Units with two level control store, a nano instruction memory is included along with memory for microinstruction. It is used to avoid redundant storing of the same operation parts of microinstructions. The operation part of microinstructions contains the address of the word in the nano instruction memory, which contains encoded control signals.

For example Microprocessors of INTEL x86 series have microprogrammed control units with a single level control store. Microprocessors Motorola 68xxx series have microprogrammed control units with two-level control stores.

Advantages of Microprogrammed Control Unit

- Compared to the Hardwired control unit, the Microprogrammed control unit has a simpler sequencing logic unit and decoders.
- It simplifies the design of the control unit. Simpler design means the control unit is cheaper and less error-prone to implement.
- It is flexible.
- It requires less chip area.

Disadvantages of Microprogrammed Control Unit

- It is slower than the Hardwired control unit.
- Microprogrammed control unit is used for CISC because it makes the design simpler and usually in CISC architecture, due to the huge number of instructions in the instruction set, the control unit is quite complex.
- It is expensive especially for small designs.

Comparison between Hardwired and Micro-programmed Control Unit

HARDWIRED CU	MICROPROGRAMMED CU
Hardware/Circuit based approach	Software/Program based approach
Implemented with the help of gates, flip flops, decoders, etc.	Microinstructions generate signals to control the execution of instructions.
Fixed instruction format.	Variable instruction format (16-64 bits per instruction).
Register-based instructions	Instructions are not register based.
ROM not used.	ROM is used.
It is used in RISC. (Reduced Instruction Set Computers)	It is used in CISC. Complex Instruction Set Computers)
decoding is faster.	decoding is slower.
Difficult to modify.	Easier to modified.
Small Chip Area	Large Chip Area

DESIGN OF THE CONTROL UNIT

A microprocessor circuit is a digital circuit which is composed of many different sequential and combinational circuits. A microprocessor unit can be divided into two main components, i.e. the Datapath and the Control unit.

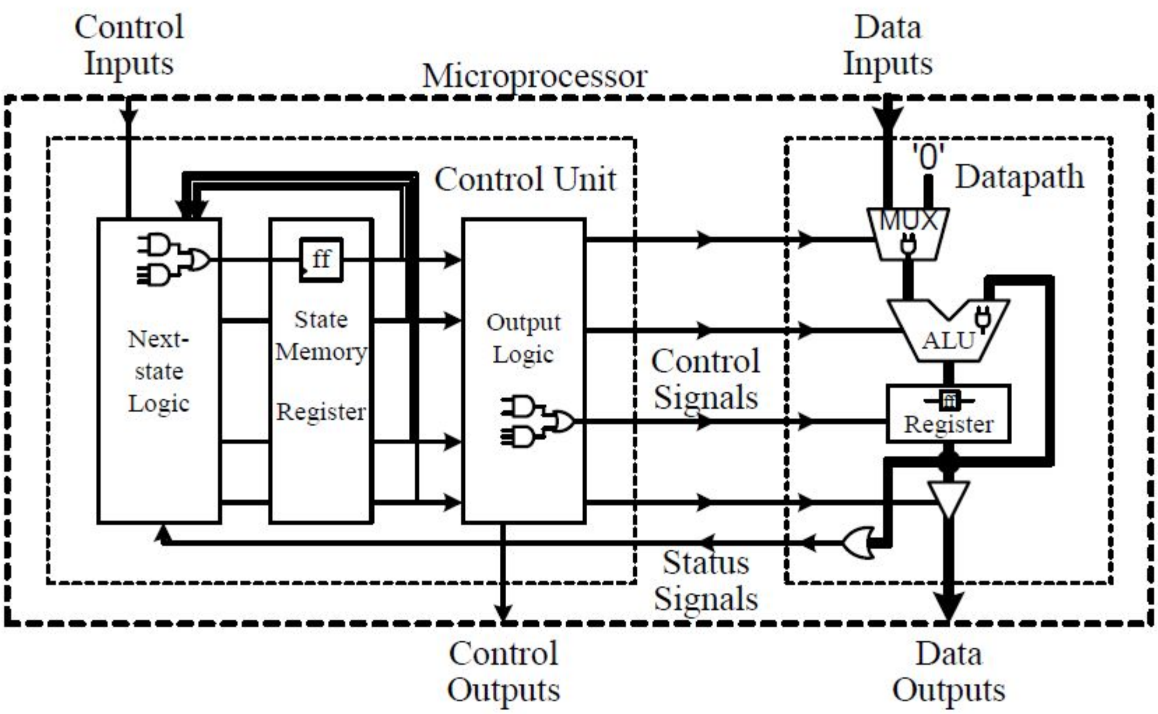


Fig : Block Diagram of a Microprocessor circuit

DATAPATH

Datapath is responsible for performing all the data manipulations required by the microprocessor. The datapath must have all of the necessary functional units to perform all the data operation needed by the microprocessor.

In general a datapath includes:

(1) functional units such as adders, shifters, multipliers, ALUs, and comparators for data manipulations,

(2) registers and other memory elements for the temporary storage of data

(3) buses, multiplexers, and tri-state buffers for the transfer of data between the different components in the datapath, and the external circuit.

CONTROL UNIT

The control unit is responsible for controlling the entire operation of the datapath, and therefore, the entire operation of the microprocessor by sending the correct control signals at the right time to the datapath. The control unit is a finite state machine consisting of three sub-components, the next-state logic, the state memory, and the output logic. Some of the control signals generated by the control unit are dependent on the data that is being manipulated within the datapath.

The state diagram for deriving the control unit is dependent on the sequence and ordering of steps involved to solve a particular problem. The control unit and the datapath are connected together via the control signals and the status signals.The Control unit generates the control signals for controlling the functional units that are inside the datapath. In return, the datapath generates status signals for the control unit in order for the control unit to make the right decision as to what to do next.

Counting Problem*: Construction of a counter microprocessor which counts from 1 to 10 and gives the count as output.*

Algorithm

```
1        i = 0
2        WHILE (i  ≠ 10) {
3             i = i + 1
4            OUTPUT i
5        }
```

Datapath design

A 4-bit-wide dedicated datapath will be constructed to generate and output the numbers from 1 to 10 for the algorithm.

For The Construction of the required datapath A 4-bit register is used for storing the value for i and an adder can be used for incrementing i.

A **Clear** signal to initialize i to 0 [Algorithm line 1]

The **iLoad** signal to load in the result from the adder, which adds a 1 to the current value of i. [Algorithm line 3]

An **Out** signal will output i. [Algorithm line 4]

The status signal for the conditional test ($i \neq 10$) is realized by the 4-input NAND gate, where the four input bits of the NAND gate are connected to the four output lines from the register as 1010 binary for the constant decimal 10.

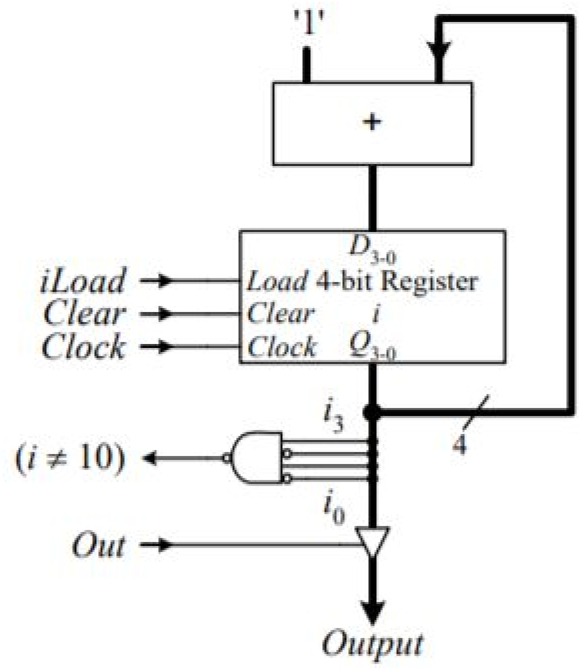


Fig : Datapath for Counting problem using a separate register and adder

Control Word	Instruction	*iLoad*	*Clear*	*Out*
1	$i = 0$	0	1	0
2	$i = i + 1$	1	0	0
3	OUTPUT i	0	0	1

Fig: Control words for solving the counting problem

Control Unit Design

The Control words are assigned to separate states of Finite state machine in the following order

S_0 : It generates the control signals for control word 1 for the execution of the instruction i = 0 .

S_1 : It generates the control signals for control word 2 for the execution of the instruction i = i + 1 .

S_2 : It generates the control signals for control word 3 for the execution of the instruction OUTPUT i .

S_3 : It marks the exit of the while loop and halting of the execution of the instruction.

The algorithm will start from **S_0** , i.e. i = 0. The next execution of the algorithm will depend on the while loop condition. The transition from **S_0** to **S_1** will depend on the condition i = 10 or not.

If **$(i \neq 10)$** , thus while condition is true , line 3 and line 4 of the algorithm will be executed resulting in state transition from **S_0** to **S_1** then **S_1** to **S_2**.

In **S_2** , if **$(i \neq 10)$**, it will loop again else it will exit the while loop resulting in transition from **S_2** to **S_3**.

The program will halt in **S_3** by having an unconditional edge going back to itself. No register transfer function is assigned to **S_3** .

Since there are 4 states in the process, 2 Flip flops will be used for construction of the control unit circuit. The following binary encoding is used for the states: (Q_1 , Q_0 represents the state of the flip flops)

State	**Q_1Q_0**
S0	00
S1	01
S2	10
S3	11

For test condition , **($i \neq 10$)** represents **True** and **($i \neq 10$)'** represents **False**.

Finite State Machine

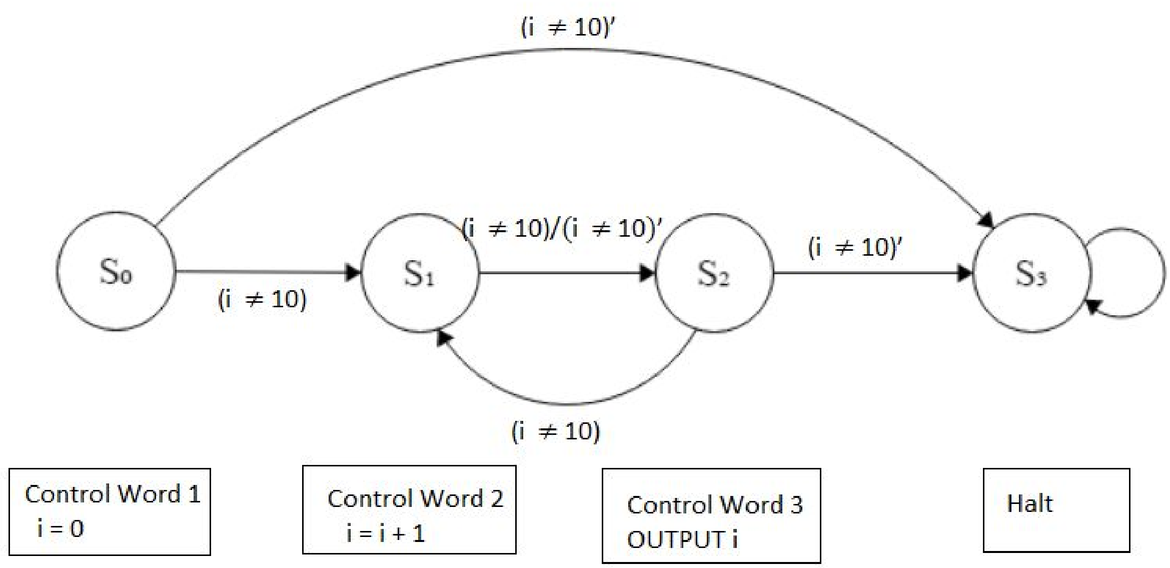


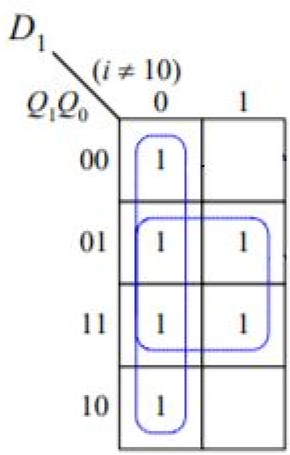
Fig : The finite state machine for the algorithm

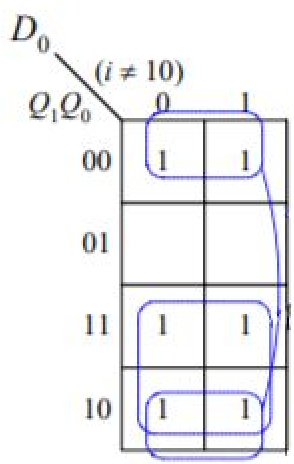
State Transition Table

Current State Q_1Q_0	Next State $Q_1^*Q_0^*$	
	$(i \neq 10)'$ [Fasle]	$(i \neq 10)$ [True]
00	11	01
01	10	10
10	11	01
11	11	11

Fig : State transition table

Solving for D_0 and D_1 with the help of K map





D1 = (i ≠ 10)′ + Q0

D0 = Q1 + Q0′

Thus the input logic for the Flip Flops are obtained.

The output logic of the final state machine is derived from the control word signals and the states in which the control words are assigned to. The control signals for controlling the operation of the datapath are simply the output signals from the output logic circuit in the Finite State Machine.

Q_1Q_0	iLoad	Clear	Out
00	0	1	0
01	1	0	0
10	0	0	1
11	0	0	0

Fig : The output logic for the control unit

Circuit Diagram of the Control Unit

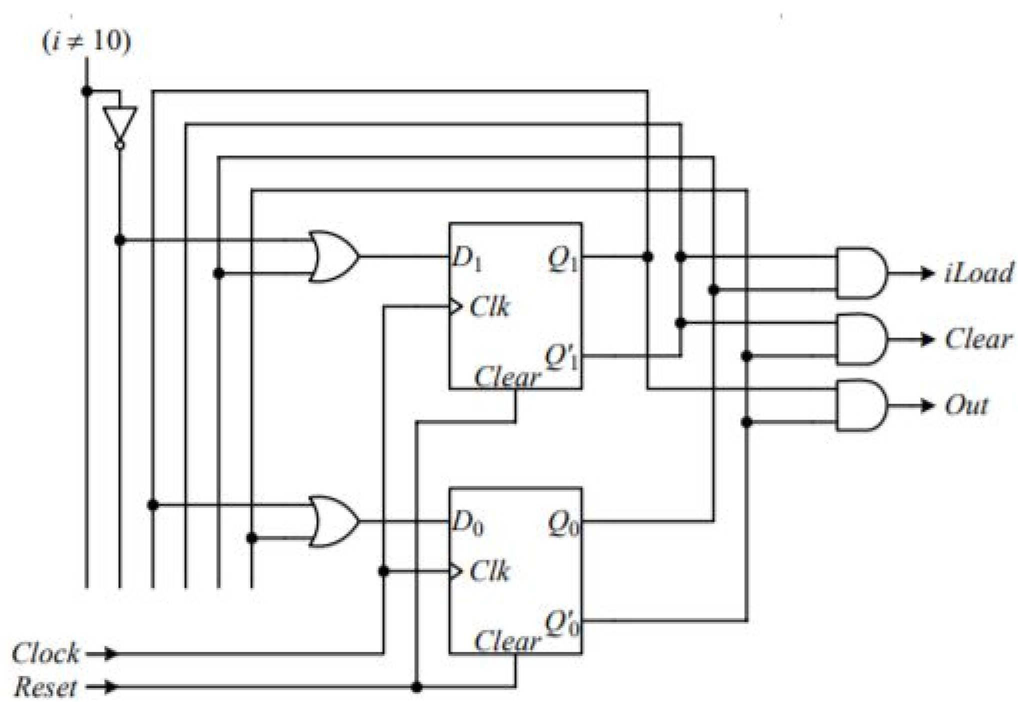


Fig : The Circuit diagram of the control unit

RESULT

The resultant circuit is a control unit for the counting problem. The state memory simply consists of the two D flip-flops with asynchronous clear signals. The two asynchronous clear signals to the flip-flops are connected to the global Reset signal so that on reset, the Finite State Machine will start from state S_0. All of the Clock signals to the flip-flops are always connected directly to the global system clock. Both the next-state logic circuit and the output logic circuit are combinational circuits and are constructed from the next-state equations and output equations, respectively. The input of the circuit is the result of the conditional statement of the while loop i.e. if ‘i’ is equal to 10 or not. The output of the circuit is the control signals for the datapath to perform the operations namely iLoad , Out and Clear in order to run the algorithm of the counting problem.

The complete report is uploaded here.[Github Repository]

IMPORTANCE OF CONTROL UNIT

Control units are an essential part of a processor that goes hand in hand in order to execute processes faster and efficiently. It manages the functionalities and timing of various processes. It directs the operation of the CPU and other units by providing timing and control signals. It also manages computer resources. It directs the flow of data between the CPU and the other devices and coordinates the activity of all other units in the computer system. Its main function is to fetch the code from the program, decode it and send the instructions to various units to perform the task.

Control unit uses many methods to keep a pipeline full and avoid stalls. Some control units do branch prediction by keeping an electronic list of the recent branches, encoded by the address of the branch instruction. Some control units do speculative execution in which computers with multiple pipelines calculate both the directions of a branch and then discard the calculations for the unused direction. The Control Unit also organises the results if several instructions are completed at the same time.

Thus a program of instructions in memory will cause the Control Unit to configure a CPU's data flows to manipulate the data correctly between instructions. This results in a computer that could run a complete program and require no human intervention to make hardware changes between instructions

CONCLUSION

Control unit is a component of a processor that coordinates the execution of different processes and manages the resources of the processor. It provides control and timing signals for the processor and other devices in the computer system. There are different kinds of control units, namely Hardwired control unit and Microprogrammed control unit. Control units are designed according to the tasks performed by it.

There are different methods of designing a control unit. For Hardwired Control unit design generally the following techniques are used:

- *State table method*: It is a simple method of sequential circuit design. Its attempt is to minimize the amount of hardware.
- *Delay element method*: It is a method which is based on the use of D flip-flop for control signal timing.
- *Sequence counter method*: It uses the counter for the timing purposes.
- *PLA method*: It uses the programmable logic array.

In this project, A simple circuit for the control unit of a counter microprocessor is designed using a state table method. The circuit generates control signals which coordinates the sequence of execution of processes according to the algorithm of the counting problem.

The final circuit obtained is the control unit for the required microprocessor which is a counter. The circuit sends the signals iLoad, Clear and Out to organise different processes in the datapath which performs the required calculations.

REFERENCE

1. Microprocessor Design Trainer Lab Manual - Enoch Huang

   http://faculty.lasierra.edu/

2. Control Unit Design

   https://www.slideshare.net/VinitRaut8/control-unit-design

3. CPU Generations

   http://btob.co.nz/business-news/five-generations-computers/

4. Microprocessor Generations , Control unit

   https://www.elprocus.com/

5. Computer Control Unit

   https://edux.pjwstk.edu.pl/