A

Project Report on

MICROCONTROLLER BASED CLAPPER SWITCH

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology

In

ELECTRONICS AND COMMUNICATION ENGINEERING

From

Rajasthan Technical University

Kota (Rajasthan)

Under the Supervision of

NEERAJ JAIN

(PROJECT GUIDE)

Submitted By:

SHOBIT AGARWAL

YASH KHANDELWAL

SANJAY KR GUPTA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

MODERN INSTITUTE OF TECHNOLOGY & RESEARCH CENTRE ALWAR (RAJ.)

SESSION: 2011-2012

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. First and Foremost we would thank the Almighty for his beautiful gift of life. The very next we would like to thank our project mentor Mr. NEERAJ JAIN, our trainer who not only guided us all through the project but also kept on giving us suggestions regarding the betterment of the project. Meanwhile, our co-trainer Sir helped so well in his absence. A very special thanks goes to our beloved Sudhanshu Sir –HOD (Dept. Of ECE) and Khushbu Sethi (Lect. ECE) who not only guided us but also provided us the morale support in due course of this project. It is his sincere motivation that has made this project possible and a success. Finally, a sincere thanks to the college management for providing us such a learning environment and resources. A special acknowledgement to all the team members for working cooperatively as a unit.

SHOBIT AGARWAL

YASH KHANDELWAL

SANJAY KR GUPTA

{GROUP MEMBERS}

B.Tech 7th Semester

PREFACE

Our journey starts with the world of embedded system .We can see a lot of examples of embedded system in our day to day life. The embedded system are exploring it self in our day today life, many of us do not know that these systems contain a processor with a lot of software embedded in it. These systems are focused to do a specific job or tasks etc. These system are now days preferred because they are easy to implement and good cost effective.

In my project with help of embedded system I had made a sound detectable clapper switch that can sense the sound with the help of a Microphone and then make a device on or off according to the previous state of that device. Sometime its better to switch on or off the devices without getting off from your position. In that condition this can be a better choice. In this project there is an implementation of both hard ware and software are present. The programming is done with help of “C” languages. The hard ware components were used to run the systems.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER – 1

CLAPPER SWITCH – AN OVERVIEW

1.1 INTRODUCTION:

I always thought the clapper so cool. For those who are lazy like me, turn on or off the bedroom light without getting out of bed, simple clap your hands is amazing. Thinking about it I projected one for me. Several projects of clappers can be found on the Internet, and most of them do not use microcontroller.

I chose to use a micro planning future updates, such as turn on or off the light with 2 claps, the TV with 3 claps, DVD player with 4 and so on. Another update can be a time to turn off the device automatically.

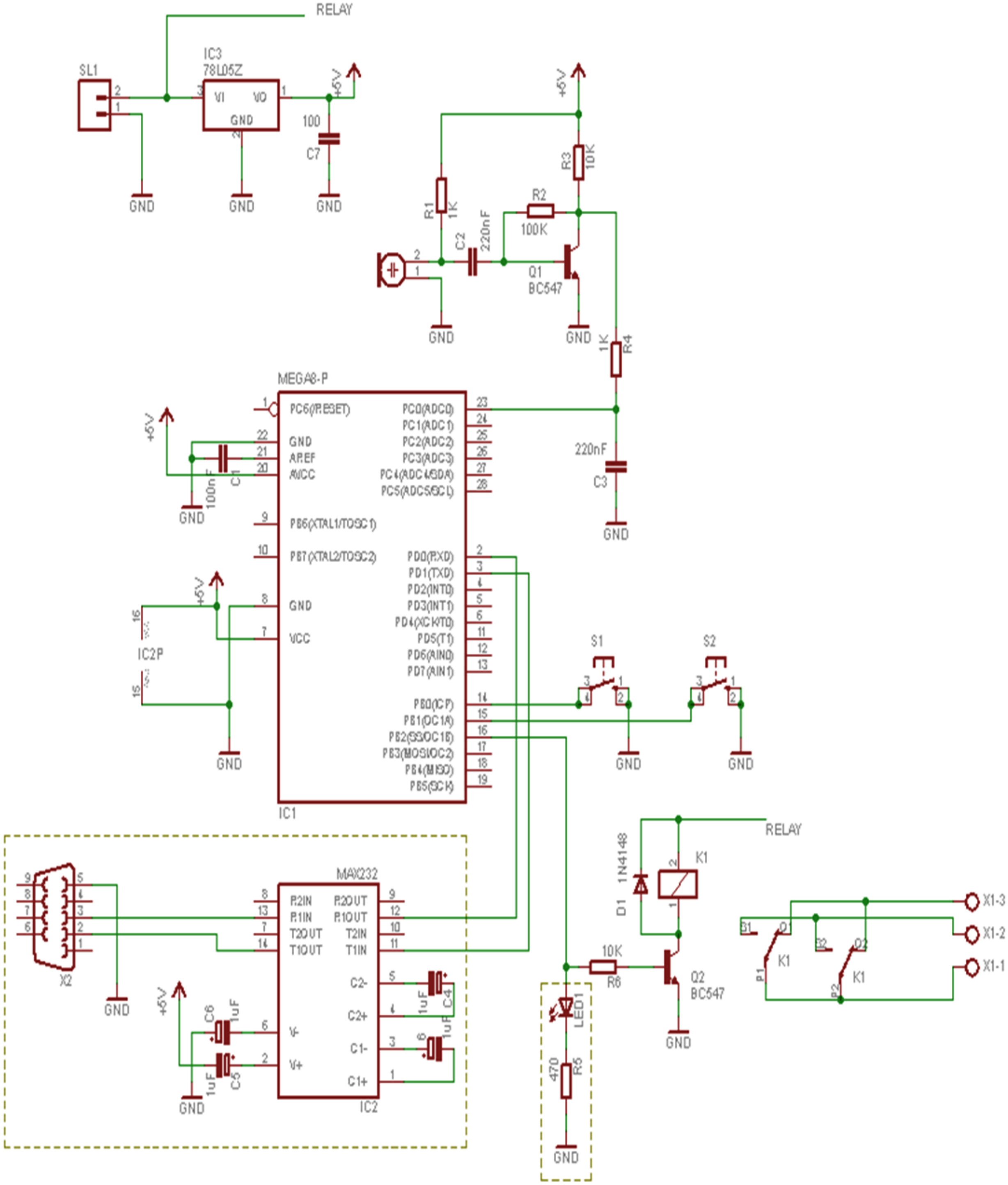
1.2 HARDWARE:

The hardware is very simple, just a microcontroller, a electret mic, a relay and some discrete components. The microcontroller is the brain of the project, I used a atmega8 because it was the only I had available.

The signal is captured by the mic and amplified by the transistor (BC548), this signal run through a lowpass filter that remove the high frequencies and go to mega8 ADC. To keep the circuit simple, only one transistor is used on signal amplification. Two buttons are used to up or down the sensibility of the signal. You can see the complete schematic below. The components on dotted line are optional.

The circuit can be powered with 7 ~ 12V and the relay must be for that voltage.

1.3 SCHEMATIC OVERVIEW:



1.4 SOFTWARE:

The software was written in C using AVR Studio and WinAVR library. The ADC is read constantly, and the returned value is compared with a variable. This variable can be changed if the user press the button S1 or S2.

So, the user can increase or decrease the intensity of the signal necessary to trigger the circuit. If the value of the ADC is less than the variable, nothing happens, otherwise the timer0 is activated and begins counting. At each timer overflow, a variable called timer is incremented.

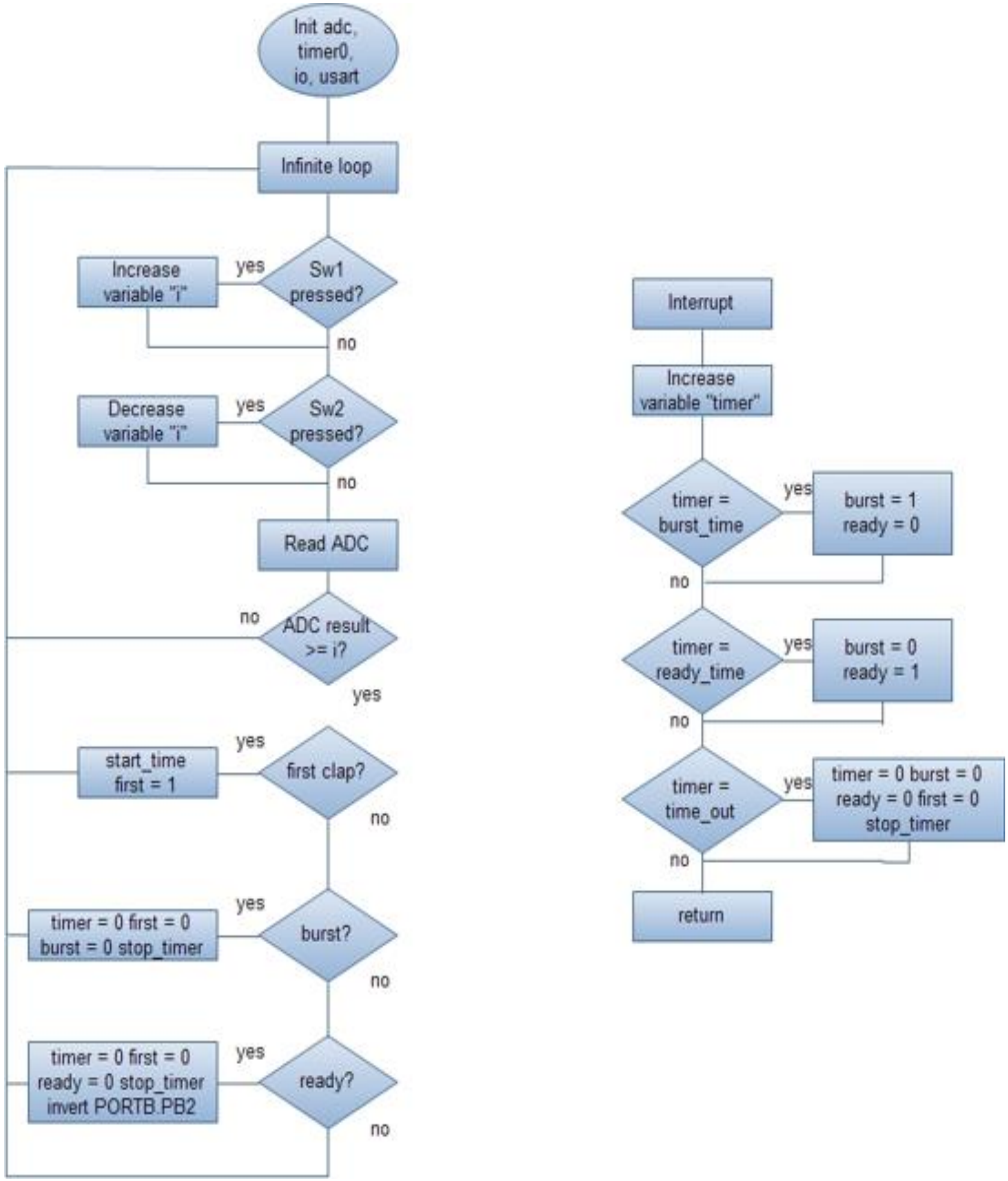
The value of variable timer is compared with three constants, burst_time, ready_time and time_out. If the value of the variable is equal to the constant burst_time the value of global variable burst become 1. If the value of timer is equal the constant ready_time the value of variable burst becomes 0 and variable ready becomes 1. If value of timer is equal time_out all the variables become equal to 0.

When another clap is detected, if burst = 1 then the microcontroller clear all variables and start the loop again. If burst = 0 and ready = 1 the output is toggled.

The relay only close when the time between a clap and another is approximately correct, thus reducing the chances of the light turn on or off accidentally.

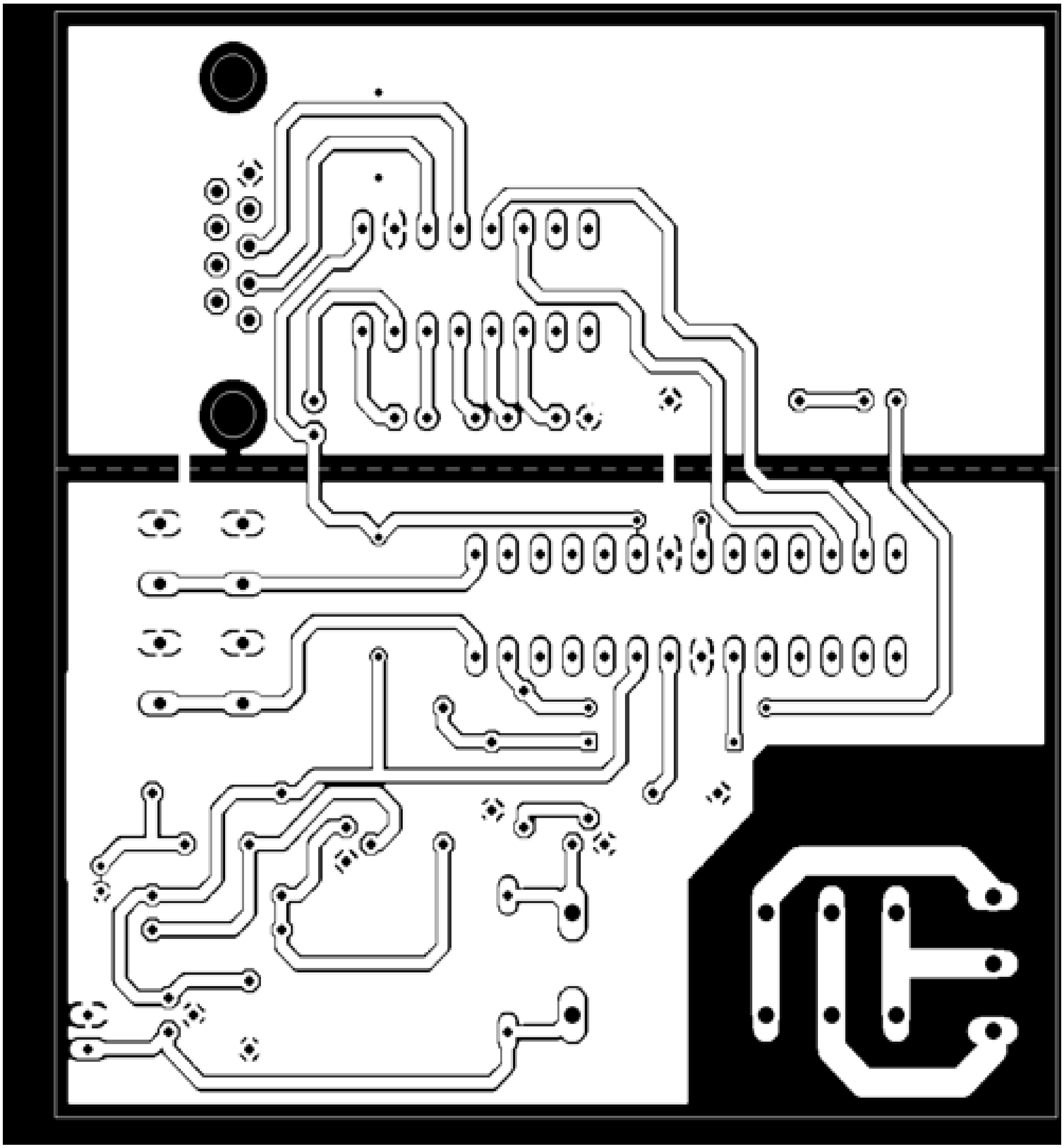
The flowchart below help understand the software

1.5 FLOWCHART FOR PROGRAM LOGIC:

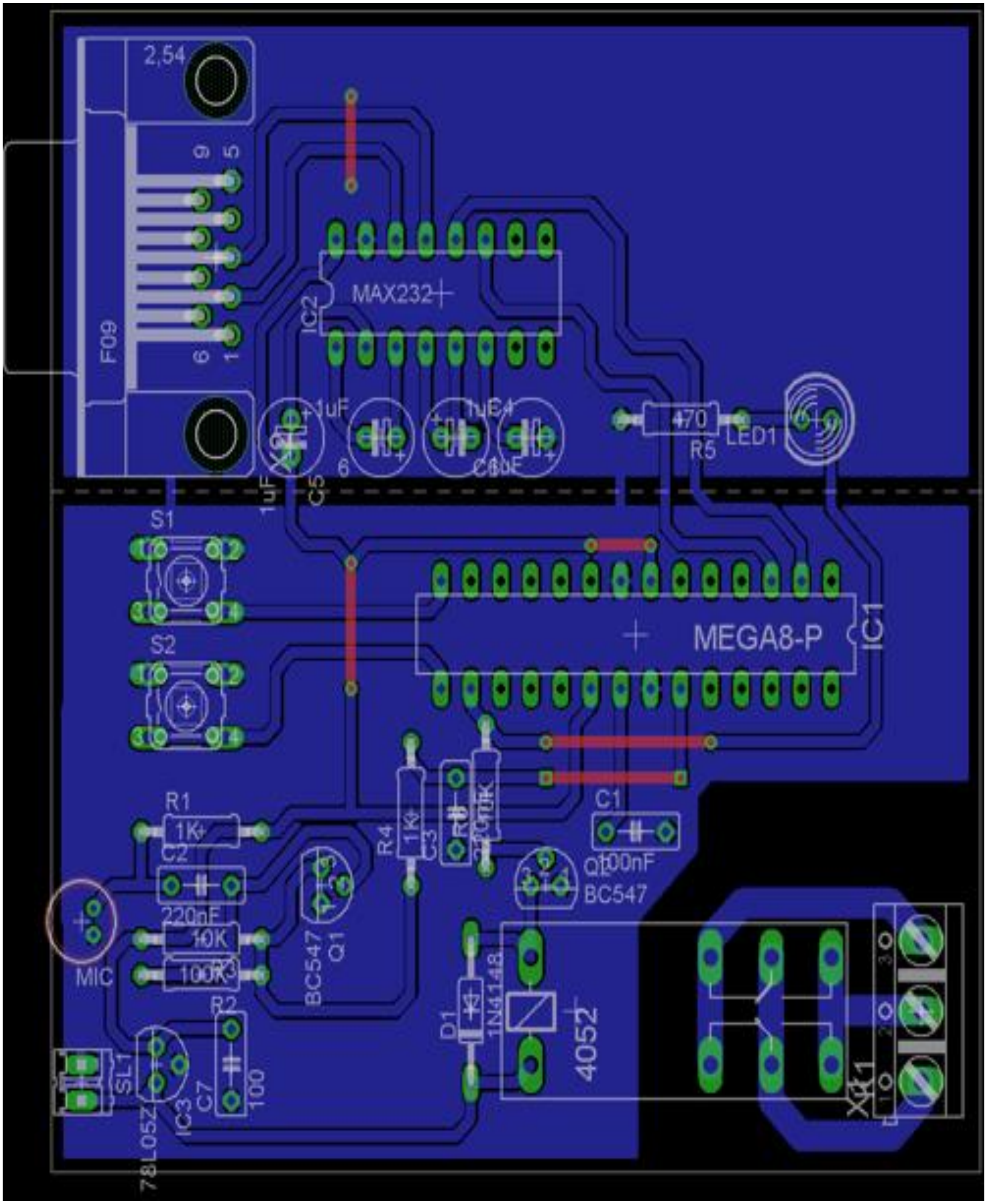


1.6 PCB LAYOUT

1.6.1 SOLDER SIDE



1.6.2 COMPONENT SIDE:



CHAPTER – 2

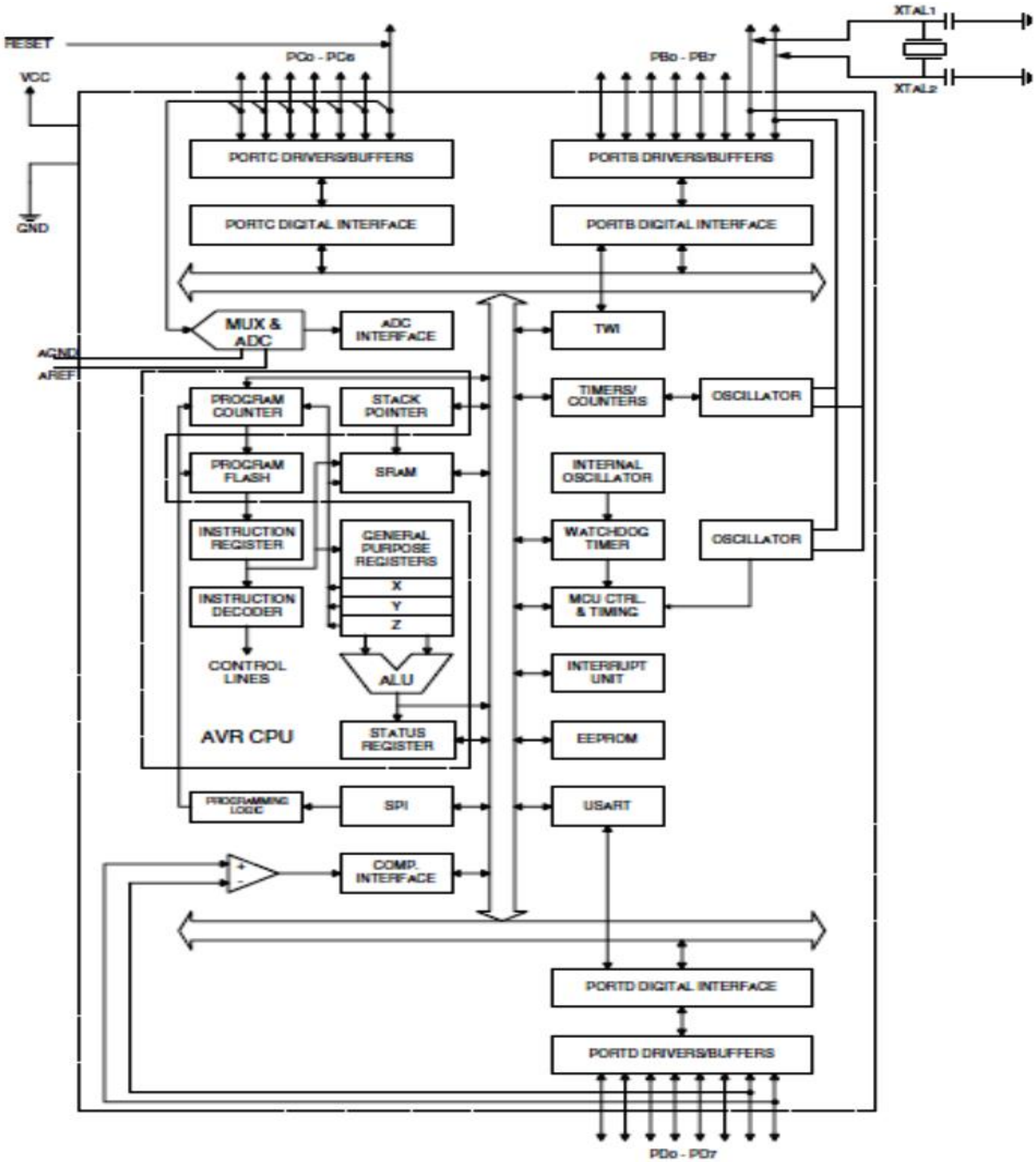
MICROCONTROLLER "ATMEGA 8-P"

2.1 MICROCONTROLLER – A BASIC APPROACH

Micro controller may be called "computer on chip", since it has basic features of microprocessor(like ALU, registers, flags, program counter, stack pointer, clock and interrupt circuit) with internal ROM, RAM, parallel and serial I/O ports within single chip. The prime use of microcontroller is to control the operation of machine using a fixed program stored in its internal ROM, which cannot be changed.

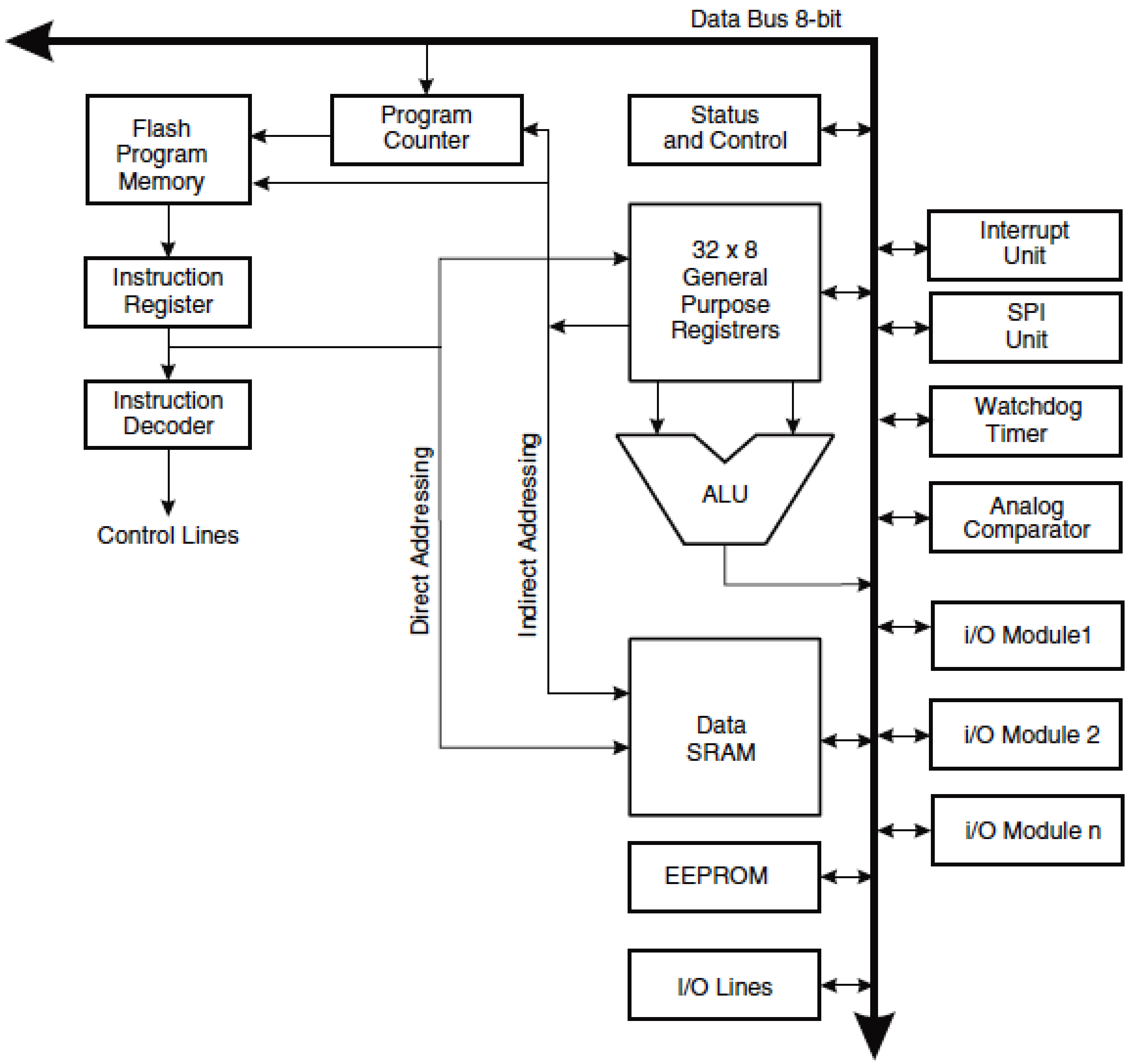
The ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed. The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. The ATmega8 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1K byte of SRAM, 23 general-purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented 2-wire Serial Interface, a 6-channel ADC (8 channels in TQFP and MLF packages) where 4 (6) channels have 10-bit accuracy and 2 channels have 8-bit accuracy, a programmable Watchdog Timer with internal oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, timer/counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction Mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. The device is manufactured using Atmel's high density nonvolatile memory technology. The Flash program memory can be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an on-chip boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash Memory. Software in the Boot Flash Section will continue to run while the Application Flash Section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.The ATmega8 AVR is supported with a full suite of program and system development tools, including C compilers, macro assemblers, program debugger/simulators, In-circuit emulators, and evaluation kits.

2.2 BLOCK DIAGRAM OF ATMEGA 8-P:



{FIG NO 2.1: BLOCK DIAGRAM OF ATMEGA 8-P}

2.3 ARCHITECTURE OVERVIEW OF "ATMEGA 8-P"



{FIG 2.2 ARCHITECTURE DETAIL OF ATMEGA 8-P}

2.3.1 Arithmetic Logic Unit – ALU:

The high-performance Atmel®AVR® ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format

2.3.2 STATUS REGISTER OF "ATMEGA 8-P":

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code. The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

{FIG 2.3 AVR STATUS REGISTER}

2.3.2.1 Bit 7 – I: Global Interrupt Enable

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions.

2.3.2.2 Bit 6 – T: Bit Copy Storage

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

2.3.2.3 Bit 5 – H: Half Carry Flag

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry is useful in BCD arithmetic.

2.3.2.4 Bit 4 – S: Sign Bit,

The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V.

2.3.2.5 Bit 3 – V: Two's Complement Overflow Flag

The Two's Complement Overflow Flag V supports two's complement arithmetics.

2.3.2.6 Bit 2 – N: Negative Flag

The Negative Flag N indicates a negative result in an arithmetic or logic operation

2.3.2.7 Bit 1 – Z: Zero Flag

The Zero Flag Z indicates a zero result in an arithmetic or logic operation.

2.3.2.8 Bit 0 – C: Carry Flag

The Carry Flag C indicates a Carry in an arithmetic or logic operation.

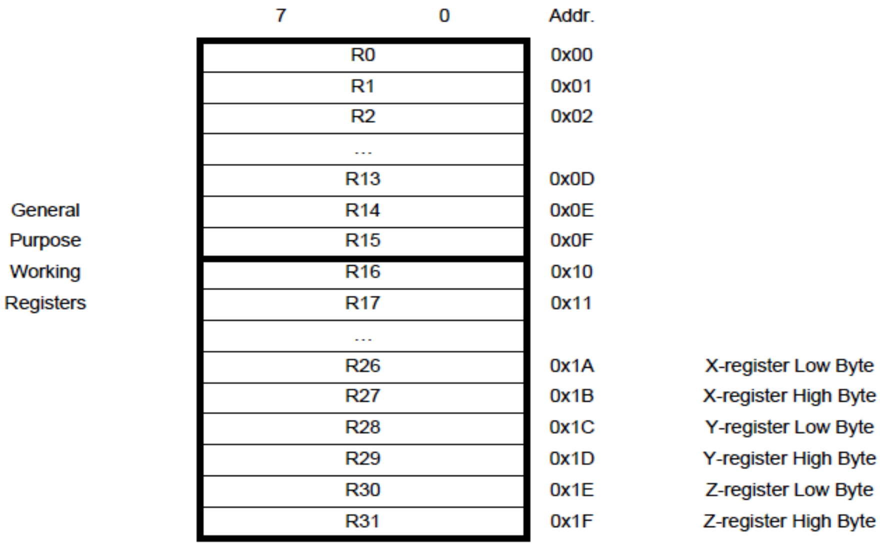
2.3.3 GENERAL PURPOSE REGISTER FILE:

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

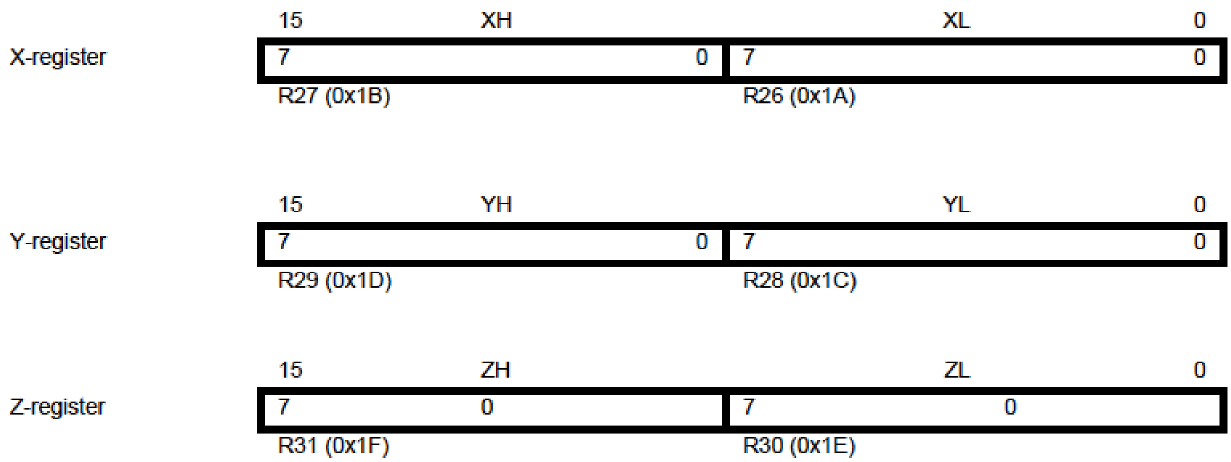
Figure 2.4 shows the structure of the 32 general purpose working registers in the CPU.

Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions. As shown in Figure 2.4, each register is also assigned a Data Memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-pointer, Y-pointer, and Z-pointer Registers can be set to index any register in the file.



{FIG 2.4 GENERAL PURPOSE REGISTER DETAIL}

2.3.4 THE X-REGISTER, Y-REGISTER AND Z-REGISTER



{FIG 2.5 ARCHITECTURE OF X,Y & Z REGISTERS}

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y and Z are defined as described in Figure 4.
In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement.

2.3.5 STACK POINTER

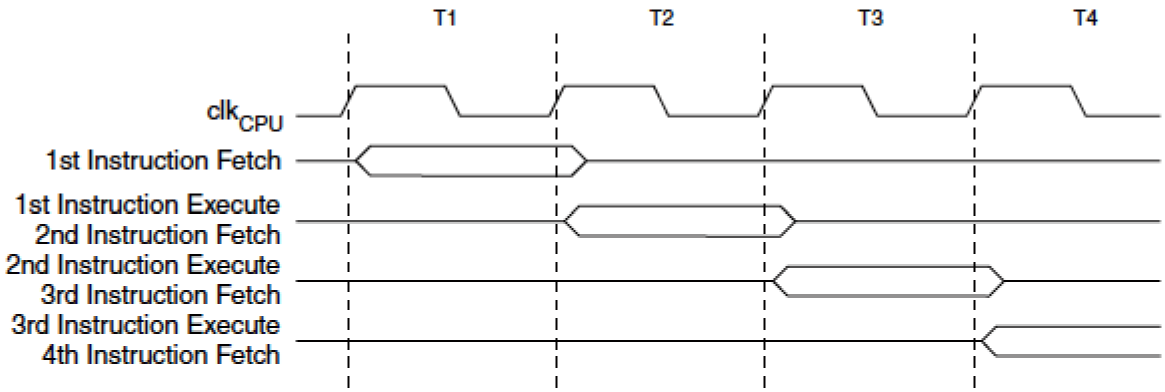
The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer. The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above 0x60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when address is popped from the Stack with return from subroutine RET or return from interrupt RETI. The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

{FIG 2.6 STACK POINTER OF ATMEGA 8-P}

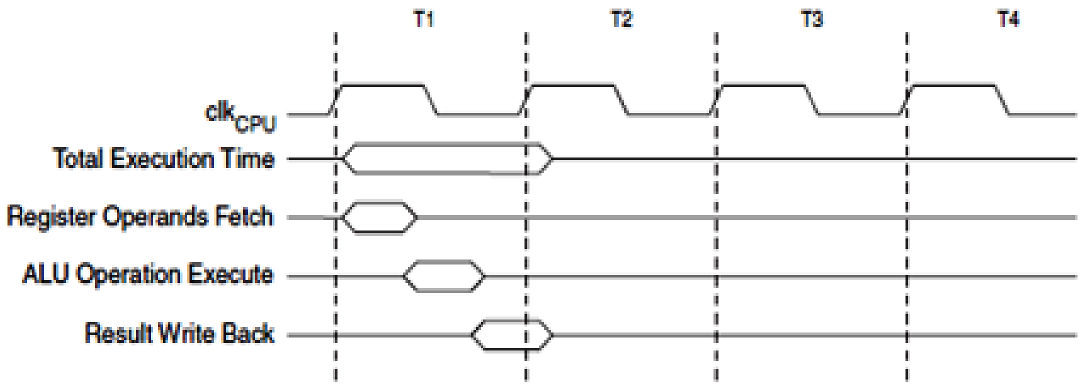
2.4 INSTRUCTION EXECUTION TIMING:

This section describes the general access timing concepts for instruction execution. The Atmel®AVR® CPU is driven by the CPU clock clkCPU, directly generated from the selected clock source for the chip. No internal clock division is used. Figure 2.7 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.



{FIG 2.7 The Parallel Instruction Fetches and Instruction Executions}

Figure 2.8 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.



{FIG 2.8 Single Cycle ALU Operation}

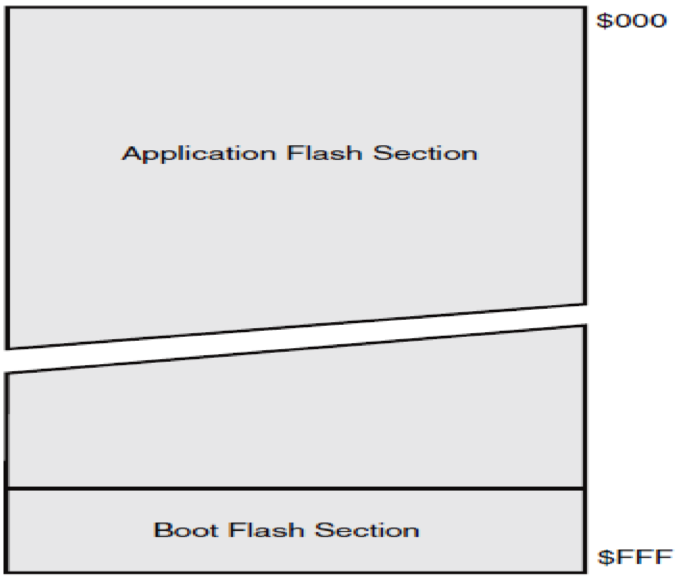
2.5 “ATMEGA 8-P” MEMORIES

This section describes the different memories in the Atmel®AVR® ATmega8. The AVR architecture has two main memory spaces, the Data memory and the Program Memory space. In addition, the ATmega8 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

2.5.1 In-System Reprogrammable Flash Program Memory

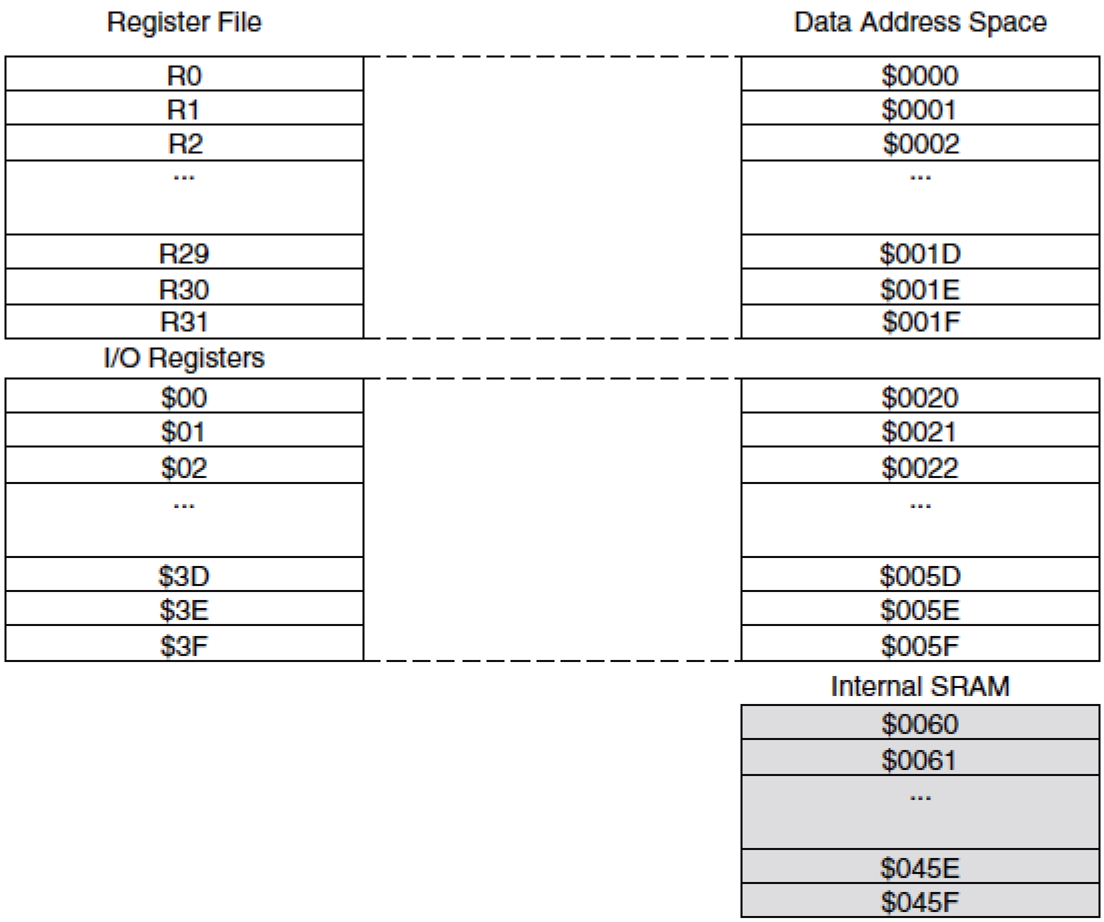
The ATmega8 contains 8Kbytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16-bits or 32-bits wide, the Flash is organized as 4K × 16 bits. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section. The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega8 Program Counter (PC) is 12 bits wide, thus addressing the 4K Program memory locations.

Figure 2.9. Program Memory Map



2.5.2 SRAM Data Memory

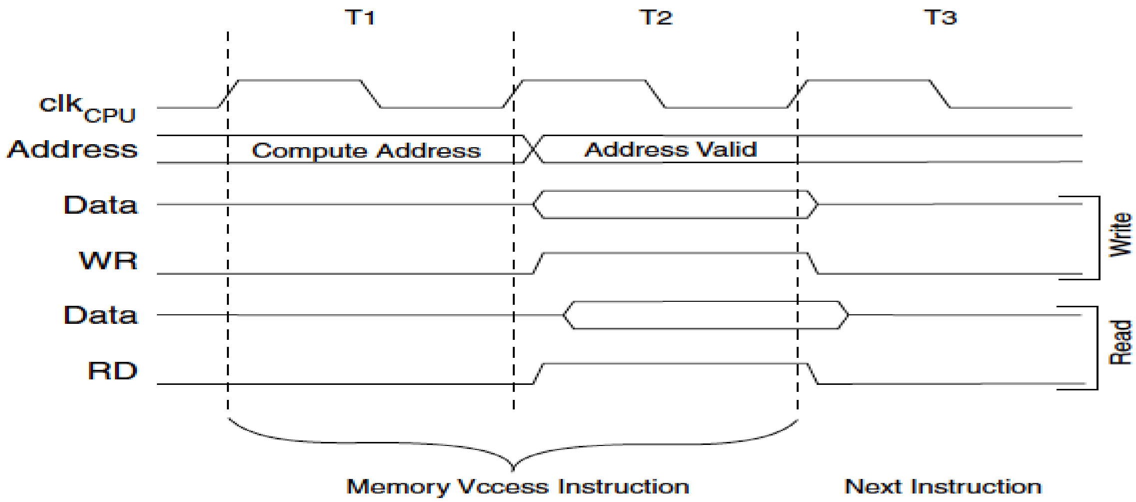
Figure 2.10 shows how the Atmel®AVR® SRAM Memory is organized. The lower 1120 Data memory locations address the Register File, the I/O Memory, and the internal data SRAM. The first 96 locations address the Register File and I/O Memory, and the next 1024 locations address the internal data SRAM. The five different addressing modes for the Data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect addressing pointer registers. The direct addressing reaches the entire data space. The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y-register or Z-register. When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y and Z are decremented or incremented. The 32 general purpose working registers, 64 I/O Registers, and the 1024 bytes of internal data SRAM in the ATmega8 are all accessible through all these addressing modes



{FIG 2.10 SRAM DATA MEMORY}

2.5.2.1 Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two clkCPU cycles as described in Figure 2.11.



{FIG 2.11 SRAM ACCESS CYCLE}

2.5.3 EEPROM Data Memory

The ATmega8 contains 512bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described below, specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

2.5.3.1 EEPROM Read/Write Access

The EEPROM Access Registers are accessible in the I/O space. The write access time for the EEPROM is given in Table 1 on page 21. A self-timing function However, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken. In heavily filtered power supplies, VCC is likely to rise or fall slowly on Power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used.

2.5.3.2 The EEPROM Address Register – EEARH and EEARL

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	–	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

{FIG 2.12 EEARH & EEARL}

- **Bits 15..9 – Res: Reserved Bits**

These bits are reserved bits in the ATmega8 and will always read as zero.

- **Bits 8..0 – EEAR8..0: EEPROM Address**

The EEPROM Address Registers – EEARH and EEARL – specify the EEPROM address in the 512bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 511. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

2.5.3.3 The EEPROM Data Register – EEDR

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

{FIG 2.13 EEDR}

- **Bits 7..0 – EEDR7..0: EEPROM Data**

For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

2.5.3.4 The EEPROM Control Register – EECR

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

{FIG 2.14 EECR}

- **7..4 – Res: Reserved Bits**

These bits are reserved bits in the Atmel®AVR® ATmega8 and will always read as zero.

- **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM Ready Interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EEWE is cleared.

- **Bit 2 – EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set, setting EEWE within four clock cycles will write data to the EEPROM at the selected address If EEMWE is zero, setting EEWE will have no effect. When EEMWE has been written to one by software, hardware clears the bit to zero after four clock cycles.

- **Bit 1 – EEWE: EEPROM Write Enable**

The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be written to one to write the value into the EEPROM. The EEMWE bit must be written to one before a logical one is written to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 3 and 4 is not essential): Wait until EEWE becomes zero Wait until SPMEN in SPMCR becomes zero Write new EEPROM address to EEAR (optional) Write new EEPROM data to EEDR (optional) Write a logical one to the EEMWE bit while writing a zero to EEWE in EECR Within four clock cycles after setting EEMWE, write a logical one to EEWE The EEPROM can not be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a boot loader allowing the CPU to program the Flash. If the Flash is never being updated by the CPU, step 2 can be omitted.

- **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR Register, the EERE bit must be written to a logic one to trigger the EEPROM read. The EEPROM read access takes one instruction, and the requested data is available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed. The user should poll the EEWE bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM, nor to change the EEAR Register. The calibrated Oscillator is used to time the EEPROM accesses. Table 1 lists the typical programming ime for EEPROM access from the CPU.

2.5.4 I/O Memory

All Atmel®AVR® ATmega8 I/Os and peripherals are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.. When using the I/O specific commands IN and OUT, the I/Oaddresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LDand ST instructions, 0x20 must be added to these addresses. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

CHAPTER – 3

THE MAX 232- DUAL DRIVER/RECIEVER

3.1 DESCRIPTION:

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5V supply. Each receiver converts TIA/EIA-232-F inputs to 5V TTL/CMOS levels. These receivers have a typical threshold of 1.3V, a typical hysteresis of 0.5 V, and can accept up to 30V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels.

3.2 FEATURES:

1. Operates from a single 5V Power Supply with 1.0uF Charge-Pump Capacitors

2. Operates up to 120 k bit/s

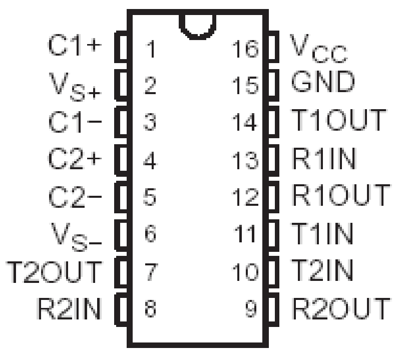
3. Two Drivers and Two Receivers

4. ±30 V Input Levels

5. Low Supply Current . . . 8 mA Typical

Applications-- TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

3.3 PIN DIAGRAM OF MAX232



{FIG 3.1 PIN LAYOUT OF MAX 232}

3.4 FUNCTION TABLE

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

H = high level, L = low level

EACH RECEIVER

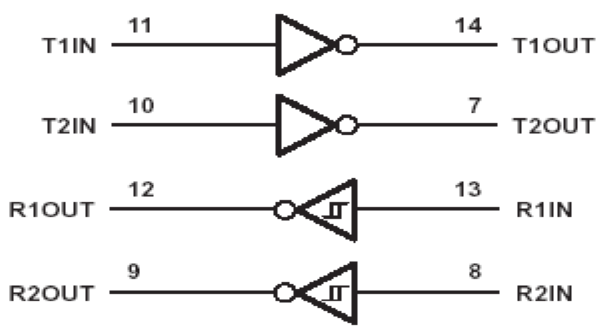
INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

{FIG 3.2 FUNCTION TABLES}

3.5 LOGIC DIAGRAM

3.5.1 POSITIVE LOGIC



{FIG 3.3 POSITIVE LOGIC}

3.6 RECOMMENDED OPERATING CONDITIONS

PARAMETER	MIN	NOR	MAX	UNIT
VCC Supply voltage	4.5	5	5.5	V
VIH High-level input voltage (T1IN,T2IN)	2			V
VIL Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN Receiver input voltage			□30	V
TA Operating free-air temperature	0		70	□□□□□C

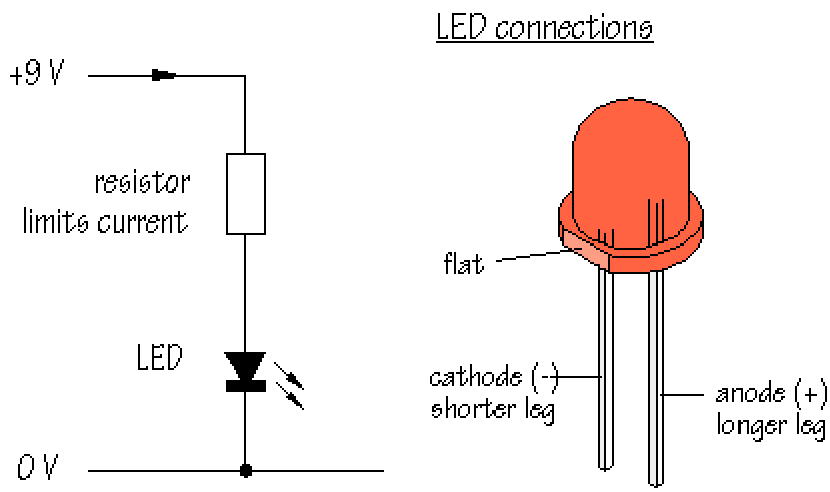
{TABLE 3.1 OPERATING CONDITIONS}

CHAPTER – 4

RESISTORS – A BRIEF THEORY

4.1 Resistor Function:-

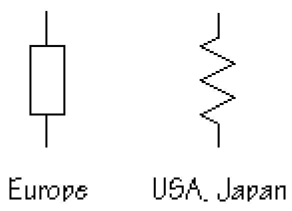
Resistors limit current. In a typical application, a resistor is connected in series with an LED:



{ FIG. 4.1 USE OF RESISTOR IN CIRCUITS}

Enough current flows to make the LED light up, but not so much that the LED is damaged. Later in this Chapter, you will find out how to calculate a suitable value for this resistor. (LEDs are described in detail in Chapter 5.)

The 'box' symbol for a fixed resistor is popular in the UK and Europe. A 'zig-zag' symbol is used in America and Japan:



{ FIG 4.2 RESISTOR SYMBOLS}

Resistors are used with transducers to make sensor subsystems. Transducers are electronic components which convert energy from one form into another, where one of the forms of energy is electrical. A light dependent resistor, or LDR, is an example of an input transducer. Changes in the brightness of the light shining onto the surface of the LDR result in changes in its resistance. As will be explained later, an input transducer is most often connected along with a resistor to to make a circuit called a potential divider. In this case, the output of the potential divider will be a voltage signal which reflects changes in illumination.

Microphones and switches are input transducers. Output transducers include loudspeakers, filament lamps and LEDs. Can you think of other examples of transducers of each type?

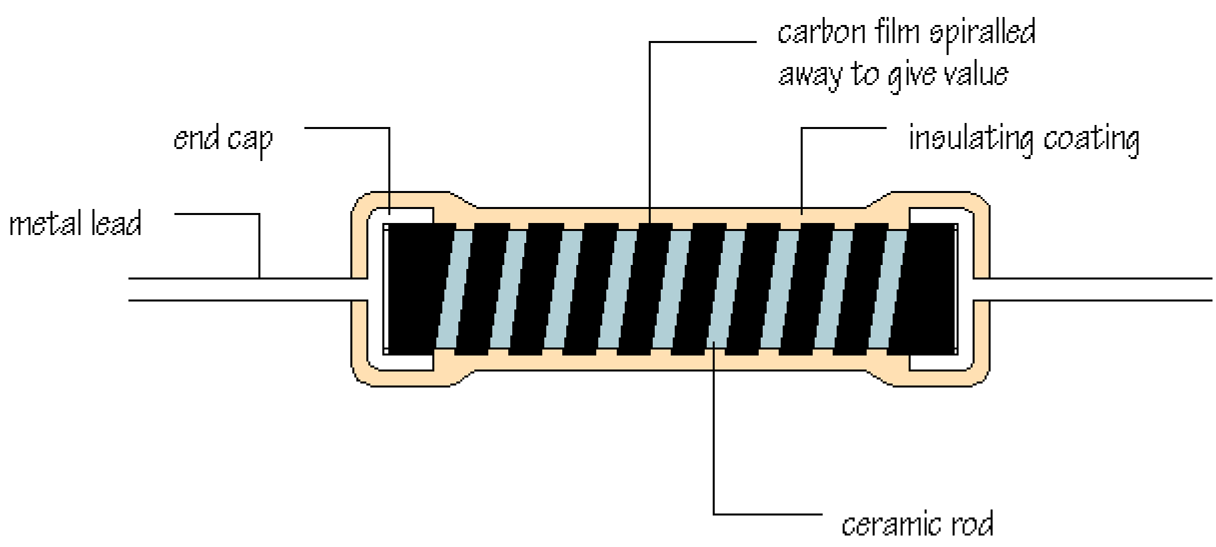
In other circuits, resistors are used to direct current flow to particular parts of the circuit, or may be used to determine the voltage gain of an amplifier. Resistors are used with capacitors (Chapter 4) to introduce time delays.

Most electronic circuits require resistors to make them work properly and it is obviously important to find out something about the different types of resistor available, and to be able to choose the correct resistor value, in Ω, kΩ, or MΩ, for a particular application.

4.2 TYPES OF RESISTORS:

4.2.1 FIXED VALUE RESISTORS:-

The diagram shows the construction of a **carbon film** resistor:



During manufacture, a thin film of carbon is deposited onto a small ceramic rod. The resistive coating is spiralled away in an automatic machine until the resistance between the two ends of the rod is as close as possible to the correct value. Metal leads and end caps are added, the resistor is covered with an insulating coating and finally painted with coloured bands to indicate the resistor value.

Carbon film resistors are cheap and easily available, with values within ±10% or ±5% of their marked, or 'nominal' value. Metal film and metal oxide resistors are made in a similar way, but can be made more accurately to within ±2% or ±1% of their nominal value. There are some differences in performance between these resistor types, but none which affect their use in simple circuits.

4.2.2 WIREWOUND RESISTORS

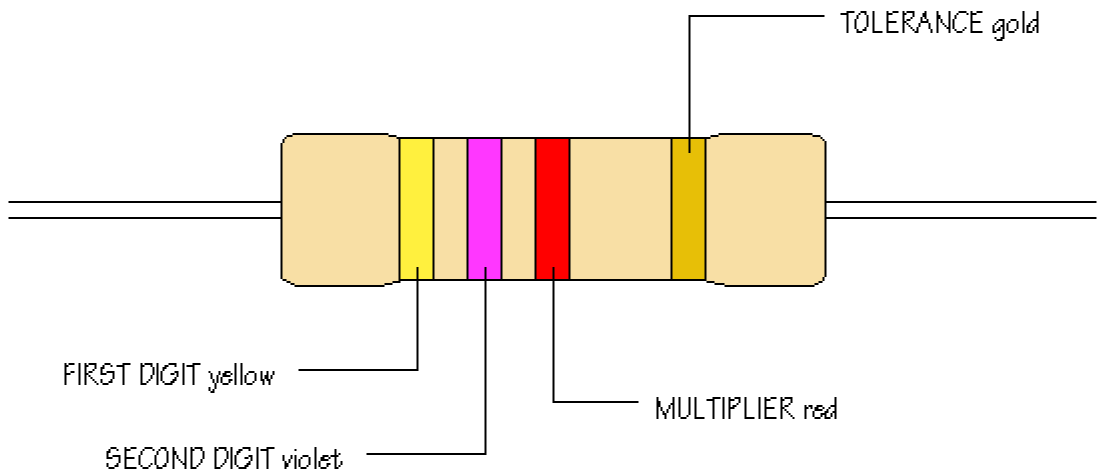
Wirewound resistors are made by winding thin wire onto a ceramic rod. They can be made extremely accurately for use in multimeters, oscilloscopes and other measuring equipment. Some types of wirewound resistors can pass large currents wihtout overheating and are used in power supplies and other high current circuits.

4.3 COLOUR CODING FOR RESISTORS:-

How can the value of a resistor be worked out from the colours of the bands? Each colour represents a number according to the following scheme:

Number	*Colour*
0	black
1	brown
2	red
3	orange
4	yellow
5	green
6	blue
7	violet
8	Grey
9	White

The first band on a resistor is interpreted as the FIRST DIGIT of the resistor value. For the resistor shown below, the first band is yellow, so the first digit is 4:



{FIG 4.4 CALCULATION USING COLOUR CODES}

The second band gives the SECOND DIGIT. This is a violet band, making the second digit 7. The third band is called the MULTIPLIER and is not interpreted in quite the same way. The multiplier tells you how many noughts you should write after the digits you already have. A red band tells you to add 2 noughts. The value of this resistor is therefore 4 7 0 0 ohms, that is, 4 700 Ω, or 4.7 $k\Omega$. Work through this example again to confirm that you understand how to apply the colour code given by the first three bands.

The remaining band is called the TOLERANCE band. This indicates the percentage accuracy of the resistor value. Most carbon film resistors have a gold-coloured tolerance band, indicating that the actual resistance value is with + or - 5% of the nominal value. Other tolerance colours are:

4.3.1 TOLERANCE OF DIFFERENT COLOURS

Tolerance	*Colour*
±1%	brown
±2%	red
±5%	gold
±10%	silver

When you want to read off a resistor value, look for the tolerance band, usually gold, and hold the resistor with the tolerance band at its right hand end. Reading resistor values quickly and accurately isn't difficult, but it does take practice!

4.3.2 CALCULATION FORMULA:

Generally hold the resistor with the tolerance band at its right hand end. Then use the following formula to calculate the value of resistor:

1st colour band gives the first digit.

2nd colour band gives the second digit.

3rd colour band gives the third digit (In case of 5 band resistors only).

4th colour band (In case of 4 band resistors it is 3rd band) is the Multiplier.

5th colour band (In case of 4 band resistors it is 4rd band) gives the Tolerance.

After observing them use

{1st digit}{2nd digit} * 10^Multiplier (+/-) Tolerance.

CHAPTER – 5

CAPACITORS – A FULL VIEW

5.1 FUNCTION :-

Capacitors store electric charge. They are used with resistors in timing circuits because it takes time for a capacitor to fill with charge. They are used to smooth varying DC supplies by acting as a reservoir of charge. They are also used in filter circuits because capacitors easily pass AC (changing) signals but they block DC (constant) signals.

5.2 CAPACITANCE:-

This is a measure of a capacitor's ability to store charge. A large capacitance means that more charge can be stored. Capacitance is measured in farads, symbol F. However 1F is very large, so prefixes are used to show the smaller values.

Three prefixes (multipliers) are used, µ (micro), n (nano) and p (pico):

- µ means 10^{-6} (millionth), so 1000000µF = 1F
- n means 10^{-9} (thousand-millionth), so 1000nF = 1µF
- p means 10^{-12} (million-millionth), so 1000pF = 1nF

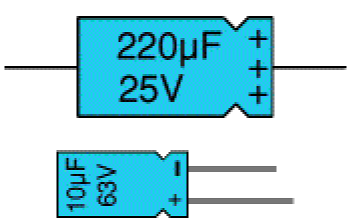
Capacitor values can be very difficult to find because there are many types of capacitor with different labelling systems!

5.3 TYPES OF CAPACITORS:-

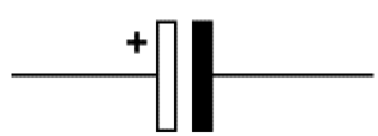
There are many types of capacitor but they can be split into two groups, polarised and unpolarised. Each group has its own circuit symbol.

5.3.1 POLARIZED CAPACITORS (LARGE VALUES, 1µF +):-

Example:- Circuit Symbol :-



:



{FIG 5-1 PLOARIZED CAPACITORS AND SYMBOL}

5.3.1.1 ELECTROLYTIC CAPACITORS:-

Electrolytic capacitors are polarised and **they must be connected the correct way round**, at least one of their leads will be marked + or -. They are not damaged by heat when soldering.

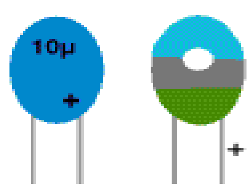
There are two designs of electrolytic capacitors; **axial** where the leads are attached to each end (220µF in picture) and **radial** where both leads are at the same end (10µF in picture). Radial capacitors tend to be a little smaller and they stand upright on the circuit board.

It is easy to find the value of electrolytic capacitors because they are clearly printed with their capacitance and voltage rating. The voltage rating can be quite low (6V for example) and it should always be checked when selecting an electrolytic capacitor. If the project parts list does not specify a voltage, choose a capacitor with a rating which is greater than the project's power supply voltage. 25V is a sensible minimum for most battery circuits.

5.3.1.2 TANTELUM BEAD CAPACITORS:-

Tantalum bead capacitors are polarised and have low voltage ratings like electrolytic capacitors. They are expensive but very small, so they are used where a large capacitance is needed in a small size.

Modern tantalum bead capacitors are printed with their capacitance, voltage and polarity in full. However older ones use a colour-code system which has two stripes (for the two digits) and a spot of colour for the number of zeros to give the value in µF. The standard colour code is used, but for the spot, **grey** is used to mean × 0.01 and **white** means × 0.1 so that values of less than 10µF can be shown. A third colour stripe near the leads shows the voltage (yellow 6.3V, black

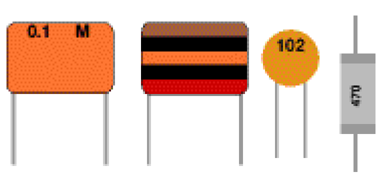


10V, green 16V, blue 20V, grey 25V, white 30V, pink 35V). The positive (+) lead is to the right when the spot is facing you: '**when the spot is in sight, the positive is to the right**'.

For example: **blue, grey, black spot** means 68µF
For example: **blue, grey, white spot** means 6.8µF
For example: **blue, grey, grey spot** means 0.68µF

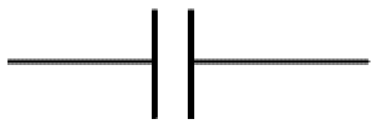
5.3.2 UNPOLARISED CAPACITORS (SMALL VALUES, up to 1µF):-

Examples:



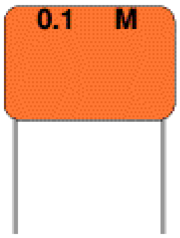
{FIG 5.2 UNPOLARISED CAPACITORS}

Circuit Symbol :



Small value capacitors are unpolarised and may be connected either way round. They are not damaged by heat when soldering, except for one unusual type (polystyrene). They have high voltage ratings of at least 50V, usually 250V or so. It can be difficult to find the values of these small capacitors because there are many types of them and several different labelling systems!

Many small value capacitors have their value printed but without a multiplier, so you need to use experience to work out what the multiplier should be!



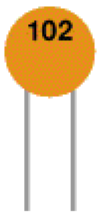
For example **0.1** means 0.1µF = 100nF.

Sometimes the multiplier is used in place of the decimal point:
For example: **4n7** means 4.7nF.

5.4 CAPACITOR NUMBER CODE:-

A number code is often used on small capacitors where printing is difficult:

- the 1st number is the 1st digit,
- the 2nd number is the 2nd digit,
- the 3rd number is the number of zeros to give the capacitance in pF.
- Ignore any letters - they just indicate tolerance and voltage rating.



For example: **102** means 1000pF = 1nF (not 102pF!)

For example: **472J** means 4700pF = 4.7nF (J means 5% tolerance).

5.5 COLOUR CODING:-

A colour code was used on polyester capacitors for many years. It is now obsolete, but of course there are many still around. The colours should be read like the resistor code, the top three colour bands giving the value in pF. Ignore the 4th band (tolerance) and 5th band (voltage rating).

For example:

brown, black, orange means 10000pF = 10nF = 0.01µF.

Note that there are no gaps between the colour bands, so 2 identical bands actually appear as a wide band.

For example:

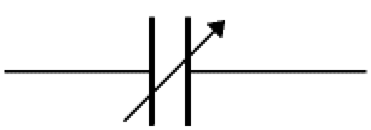
wide red, yellow means 220nF = 0.22µF.

Colour Code

Colour	Number
Black	0
Brown	1
Red	2
Orange	3
Yellow	4
Green	5
Blue	6
Violet	7
Grey	8
White	9

{TABLE 5.1 COLOUR CODING}

5.6 Variable capacitors:-



{FIG 5.3 Variable Capacitor Symbol}

Variable capacitors are mostly used in radio tuning circuits and they are sometimes called 'tuning capacitors'. They have very small capacitance values, typically between 100pF and 500pF (100pF = 0.0001µF). The type illustrated usually has trimmers built in (for making small adjustments - see below) as well as the main variable capacitor.

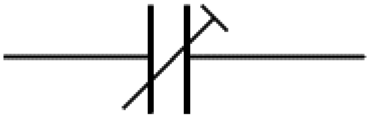
Variable Capacitor

Many variable capacitors have very short spindles which are not suitable for the standard knobs used for variable resistors and rotary switches. It would be wise to check that a suitable knob is available before ordering a variable capacitor.

Variable capacitors are **not** normally used in timing circuits because their capacitance is too small to be practical and the range of values available is very limited. Instead timing circuits use a fixed capacitor and a variable resistor if it is necessary to vary the time period.

5.7 TRIMMER CAPACITORS:-

Trimmer capacitors (trimmers) are miniature variable capacitors. They are designed to be mounted directly onto the circuit board and adjusted only when the circuit is built.



Trimmer Capacitor Symbol

A small screwdriver or similar tool is required to adjust trimmers. The process of adjusting them requires patience because the presence of your hand and the tool will slightly change the capacitance of the circuit in the region of the trimmer!

Trimmer capacitors are only available with very small capacitances, normally less than 100pF. It is impossible to reduce their capacitance to zero, so they are usually specified by their minimum and maximum values, for example 2-10pF.

Trimmer Capacitor

CHAPTER – 6

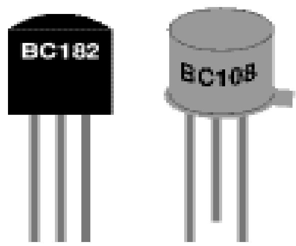
TRANSISTORS – A CONCEPTUAL VIEW

6.1 FUNCTION:-

Transistors amplify current, for example they can be used to amplify the small output current from a logic IC so that it can operate a lamp, relay or other high current device. In many circuits a resistor is used to convert the changing current to a changing voltage, so the transistor is being used to amplify voltage.

A transistor may be used as a switch (either fully on with maximum current, or fully off with no current) and as an amplifier(always partly on).

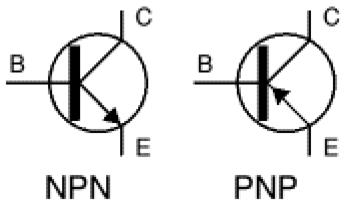
The amount of current amplification is called the current gain, symbol h_{FE}.



{FIG 6.1 EXAMPLES OF TRANSISTORS}

6.2 TYPES OF TRNASISTORS:-

There are two types of standard transistors, NPN and PNP, with different circuit symbols. The letters refer to the layers of semiconductor material used to make the transistor. Most transistors used today are NPN because this is the easiest type to make from silicon. If you are new to electronics it is best to start by learning how to use NPN transistors.



{FIG 6.2Transistor circuit symbols}

The leads are labelled base (B), collector (C) and emitter (E).
These terms refer to the internal operation of a transistor but they are not much help in understanding how a transistor is used, so just treat them as labels!

A Darlington pair is two transistors connected together to give a very high current gain.

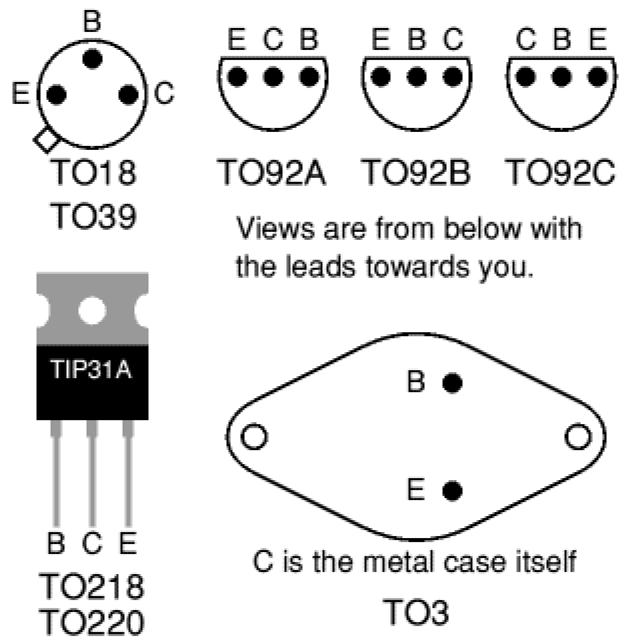
6.3 CONNECTING:-

Transistors have three leads which must be connected the correct way round. Please take care with this because a wrongly connected transistor may be damaged instantly when you switch on.

If you are lucky the orientation of the transistor will be clear from the PCB or stripboard layout diagram, otherwise you will need to refer to a supplier's catalogue to identify the leads.

The drawings on the right show the leads for some of the most common case styles.

Please note that transistor lead diagrams show the view from below with the leads towards you. This is the opposite of IC (chip) pin diagrams which show the view from above.



{FIG 6.3Transistor leads for some common case styles.}

In addition to standard (bipolar junction) transistors, there are field-effect transistors which are usually referred to as FETs. They have different circuit symbols and properties and they are not (yet) covered by this page.

6.4 FUNCTIONAL MODEL OF NPN TRANSISTOR:-

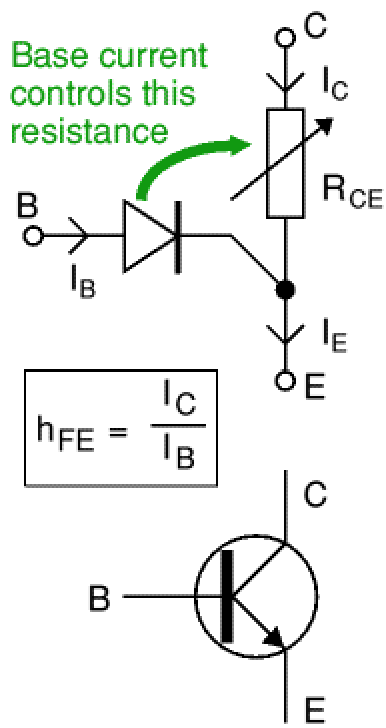
The operation of a transistor is difficult to explain and understand in terms of its internal structure. It is more helpful to use this functional model:

- The base-emitter junction behaves like a diode.
- A base current I_B flows only when the voltage V_{BE} across the base-emitter junction is 0.7V or more.
- The small base current I_B controls the large collector current Ic.
- $Ic = h_{FE} \times I_B$ (unless the transistor is full on and saturated)
  h_{FE} is the current gain (strictly the DC current gain), a typical value for h_{FE} is 100 (it has no units because it is a ratio)
- The collector-emitter resistance R_{CE} is controlled by the base current I_B:
  - $I_B = 0$ R_{CE} = infinity transistor off
  - I_B small R_{CE} reduced transistor partly on
  - I_B increased $R_{CE} = 0$ transistor full on ('saturated')

6.5 ADDITIONAL NOTES:

- A resistor is often needed in series with the base connection to limit the base current I_B and prevent the transistor being damaged.
- Transistors have a maximum collector current Ic rating.
- The **current gain h_{FE} can vary widely**, even for transistors of the same type!
- A transistor that is **full on** (with R_{CE} = 0) is said to be '**saturated**'.
- When a transistor is saturated the collector-emitter voltage V_{CE} is reduced to almost 0V.
- When a transistor is saturated the collector current Ic is determined by the supply voltage and the external resistance in the collector circuit, not by the transistor's current gain. As a result the ratio Ic/I_B for a saturated transistor is less than the current gain h_{FE}.
- The emitter current I_E = Ic + I_B, but Ic is much larger than I_B, so roughly I_E = Ic.

There is a table showing technical data for some popular transistors on the transistors page.



6.6 TRANSISTOR TESTING:-

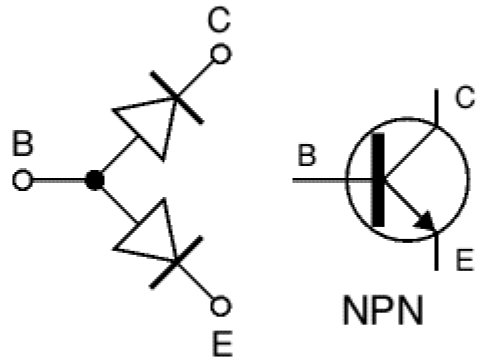
Transistors can be damaged by heat when soldering or by misuse in a circuit. If you suspect that a transistor may be damaged there are two easy ways to test it:

6.6.1 TESTING WITH MULTIMETER:-

Use a multimeter or a simple tester (battery, resistor and LED) to check each pair of leads for conduction. Set a digital multimeter to diode test and an analogue multimeter to a low resistance range.

Test each pair of leads both ways (six tests in total):

- The base-emitter (BE) junction should behave like a diode and conduct one way only.
- The base-collector (BC) junction should behave like a diode and conduct one way only.
- The collector-emitter (CE) should not conduct either way.



{FIG 6.4Testing an NPN transistor}

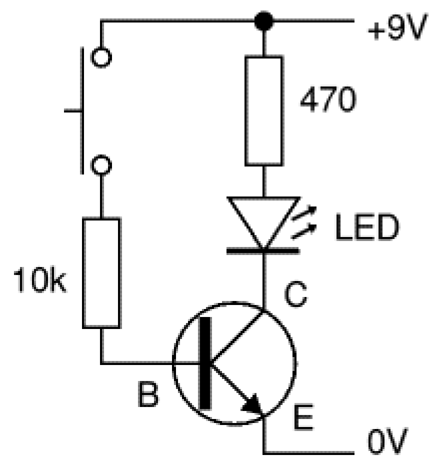
The diagram shows how the junctions behave in an NPN transistor. The diodes are reversed in a PNP transistor but the same test procedure can be used.

6.6.2.TESTING IN SIMPLE SWITCHING CIRCUIT:-

Connect the transistor into the circuit shown on the right which uses the transistor as a switch. The supply voltage is not critical, anything between 5 and 12V is suitable. This circuit can be quickly built on breadboardfor example. Take care to include the 10kΩ resistor in the base connection or you will destroy the transistor as you test it!

If the transistor is OK the LED should light when the switch is pressed and not light when the switch is released.

To test a PNP transistor use the same circuit but reverse the LED and the supply voltage.

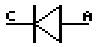


{FIG 6.5 A simple switching circuit to test an NPN transistor}

Some multimeters have a 'transistor test' function which provides a known base current and measures the collector current so as to display the transistor's DC current gain h_{FE}.

CHAPTER – 7

DIODE – A BASIC CONCEPT

7.1 DIODE  :-

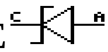
A two-terminal semiconductor (rectifying) device that exhibits a nonlinear current-voltage characteristic. The function of a diode is to allow current in one direction and to block current in the opposite direction. The terminals of a diode are called the anode and cathode. There are two kinds of semiconductor diodes: a P-N junction diode, which forms an electrical barrier at the interface between N- and P-type semiconductor layers, and a Schottky diode, whose barrier is formed between metal and semiconductor regions.

But this discussion really ought to start with a bit about semiconductors as materials.

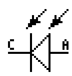
Semiconductors are crystals that, in their pure state, are resistive (that is, their electrical properties lie between those of conductors and insulators) -- but when the proper impurities are added (this process is called doping) in trace amounts (often measured in parts per billion), display interesting and useful properties.

7.2 A BIT OF HISTORY:-

The oldest ancestor of semiconductor devices was the crystal detector, used in early wireless radios. This device (patented by a German scientist, Ferdinand Braun, in 1899) was made of a single metal wire (fondly called a "cat's whisker") touching against a semiconductor crystal. The result was a "rectifying diode" (so called because it has two terminals), which lets current through easily one way, but hinders flow the other way. By 1930, though, vacuum-tube diodes had all but replaced the smaller but much quirkier crystal detector. The crystal and "cat's whisker" were left to languish as a kids' toy in the form of "crystal radios." The development of radar during World War II did much to revive the fortunes of crystal detectors (and, as a result, that of semiconductors) -- although temperamental, crystals were better than vacuum-tube diodes at rectifying the high frequencies used by radar. So, during the war, much effort was put into improving the semiconductors, mostly silicon and germanium, used in crystal detectors. At about the same time, Russell Ohl at Bell Laboratories discovered that these materials could be "doped" with small amounts of foreign "impurity" atoms to create interesting new properties. Depending on the selection of impurities (often called dopants) added, semiconductor material of two electricallly-different types can be created -- one that iselectron-rich (called N-type, where N stands for Negative), or one that is electron-poor (called P-type, where P stands for Positive). Most of the "magic" ofsemiconductor devices occurs at the boundary between P-type and N-type semiconductor material -- such a boundary is called a P-N junction. Ohl and his colleagues found that such a P-N junction made an effective diode. Like many components, diodes have a positive side or leg (a.k.a, their anode), and a negative side (cathode). When the voltage on the anode is higher than on thecathode then current flows through the diode (the resistance is very low). When the voltage is lower on the anode than on the cathode then the current does not flow (the resistance is very high). An easy way to remember this is to look at the symbol for a diode -- the "arrow" in the diode symbol points the direction in which it allows current (hole flow) to flow. The cathode of a diode is generally marked with a line next to it (on the diode body). You can see a similar line in the schematic symbols, above. Diodes are also some times marked with an identifying color code (similar, but not identical, to that used for resistors); a good explanation is given here. Note that when current is flowing through a diode, the voltage on the positive leg is higher than on the negative leg (this is often referred to as the diode's "forward voltage drop"). The magnitude of the voltage drop is a function of (among other things) the semiconductor material that the diode is made from. Silicon diodes are the most common and cheapest, and have a forward voltage drop of about 0.65 volts. Germanium diodes have a forward voltage drop of about 0.1 volt. Germanium diodes, though, are typically much more expensive than silicon diodes; luckily, they're salvageable from lots of circuit boards.

7.3 ZENER DIODE  :-

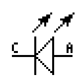
The Zener diode is designed to have a specific reverse breakdown voltage (i.e., conduction voltage when reverse-biased). Because of this, Zener diodes can be used by themselves as voltage-sensitive switches, or in series with a current-limiting resistor to provide voltage regulation.

7.4 PHOTODIODES  :-

All P-N junctions are light sensitive; photodiodes are just P-N junctions that are designed to optimize this effect. Photodiodes can be used two ways -- in a photovoltaic (here it becomes a current source when illuminated -- see solar cell), or photoconductive role.

- To use a photodiode in its photoconductive mode, the photodiode is reverse-biased; the photodiode will then allow a current to flow when it is illuminated.

  ThermoCentrovision has an interesting site on the technology behind photodiodes here.

7.4.1 LIGHT EMITTING DIODES (LEDs)  :-

All diodes emit some light when forward-biased. LEDs are made from a special semiconductor (like gallium arsenide phosphide) which optimizes this light output. Unlike light bulbs, LEDs rarely burn out unless their current limit is passed.

When current is flowing through an LED the voltage on the positive leg is about 1.4 volts higher than the voltage on the negative side (this varies with LED type -- infrared LEDs have a lower forward voltage requirement, others may need up to 1.8 V). Remember that there is very little resistance to limit the current, so a resistor must be used in series with the LED to avoid destroying it (note, though, that some panel-mount LEDs come from the factory with a current-limiting resistor soldered to them).

Also note that LEDs can be used as photodiodes (tho' their sensitivity is relatively low, so they're only useable this way in very bright conditions).

7.4.2 FLASHING LEDs (FLEDs):-

A flashing LED is just an LED with a built-in microcircuit to cause it to flash periodically. Since the FLED draws current when it flashes, we can use FLEDs to drive a number of timing-dependent circuits (via the fact that it periodically becomes conductive). In particular, see the discussion of the

FLED solarenginedesign. Like other LEDs, FLEDs are light-sensitive, and so flash faster in brighter light. Note that some FLEDs need 3 V minimum to work in, but FLEDs don't in general require current-limiting resistors (at least, I've never seen one that does).

7.5 CURRENT VOLTAGE CHARTERISTICS:-

A semiconductor diode's current–voltage characteristic, or I–V curve, is related to the transport of carriers through the so-called depletion layeror depletion region that exists at the p-n junction between differing semiconductors. When a p-n junction is first created, conduction band (mobile) electrons from the N-doped region diffuse into the P-doped region where there is a large population of holes (places for electrons in which no electron is present) with which the electrons "recombine". When a mobile electron recombines with a hole, both hole and electron vanish, leaving behind an immobile positively charged donor (the dopant) on the N-side and negatively charged acceptor (the dopant) on the P-side. The region around the p-n junction becomes depleted of charge carriers and thus behaves as an insulator. However, the width of the depletion region (called the depletion width) cannot grow without limit. For each electron-hole pair that recombines, a positively-charged dopant ion is left behind in the N-doped region, and a negatively charged dopant ion is left behind in the P-doped region. As recombination proceeds and more ions are created, an increasing electric field develops through the depletion zone which acts to slow and then finally stop recombination. At this point, there is a "built-in" potential across the depletion zone. If an external voltage is placed across the diode with the same polarity as the built-in potential, the depletion zone continues to act as an insulator, preventing any significant electric current flow (unless electron/hole pairs are actively being created in the junction by, for instance, light. see photodiode). This is the reverse bias phenomenon. However, if the polarity of the external voltage opposes the built-in potential, recombination can once again proceed, resulting in substantial electric current through the p-n junction (i.e. substantial numbers of electrons and holes recombine at the junction).. For silicon diodes, the built-in potential is approximately 0.6 V. Thus, if an external current is passed through the diode, about 0.6 V will be developed across the diode such that the P-doped region is positive with respect to the N-doped region and the diode is said to be "turned on" as it has a forward bias.

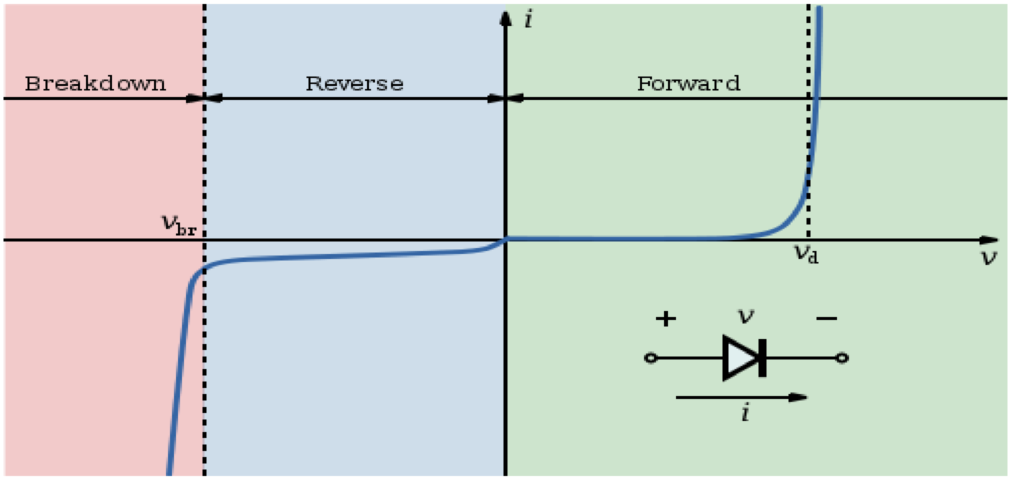


Figure 7.1: I–V characteristics of a P-N junction diode (not to scale).

A diode’s I–V characteristic can be approximated by four regions of operation (see the figure at right). At very large reverse bias, beyond the peak inverse voltage or PIV, a process called reverse breakdownoccurs which causes a large increase in current (i.e. a large number of electrons and holes are created at, and move away from the pn junction) that usually damages the device permanently. Theavalanche diode is deliberately designed for use in the avalanche region. In the zener diode, the concept of PIV is not applicable. A zener diode contains a heavily doped p-n junction allowing electrons to tunnel from the valence band of the p-type material to the conduction band of the n-type material, such that the reverse voltage is "clamped" to a known value (called the zener voltage), and avalanche does not occur. Both devices, however, do have a limit to the maximum current and power in the clamped reverse voltage region. Also, following the end of forward conduction in any diode, there is reverse current for a short time. The device does not attain its full blocking capability until the reverse current ceases. The second region, at reverse biases more positive than the PIV, has only a very small reverse saturation current. In the reverse bias region for a normal P-N rectifier diode, the current through the device is very low (in the µA range). However, this is temperature dependent, and at suffiently high temperatures, a substantial amount of reverse current can be observed (mA or more). The third region is forward but small bias, where only a small forward current is conducted. As the potential difference is increased above an arbitrarily defined "cut-in voltage" or "on-voltage" or "diode forward voltage drop (V_d)", the diode current becomes appreciable (the level of current considered "appreciable" and the value of cut-in voltage depends on the application), and the diode presents a very low resistance. The current–voltage curve is exponential. In a normal silicon diode at rated currents, the arbitrary "cut-in" voltage is defined as 0.6 to 0.7 volts. The value is different for other diode types — Schottky diodes can be as low as 0.2 V and red light-emitting diodes (LEDs) can be 1.4 V or more and blue LEDs can be up to 4.0 V. At higher currents the forward voltage drop of the diode increases. A drop of 1 V to 1.5 V is typical at full rated current for power diodes.

7.6 DIODE DETECTOR CIRCUIT:-

A two-electrode tube or diode can be used as a detector with much more satisfactory results than the crystal just described.

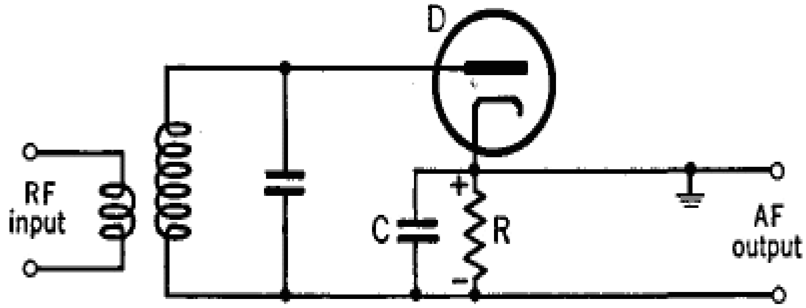


Fig. 7.2. A simple diode detector circuit

In the circuit of Fig. 17 C, the r.f. current is rectified by the diodeD and flows through the load resistance R. Since (in the conventional sense) current can only flow from the plate to the filament of the tube, the voltage drop across R is + at the top and — at the bottom, as shown. This voltage varies in strength and frequency in the same way as the modulations of the modulated carrier wave or r.f. input. A typical value of R is 250,000 ohms. Condenser C must have a reactance for the given r.f. which is small compared with the resistance of R. If C is too large some of the a.f. will be lost. A typical value is 250 μμf.

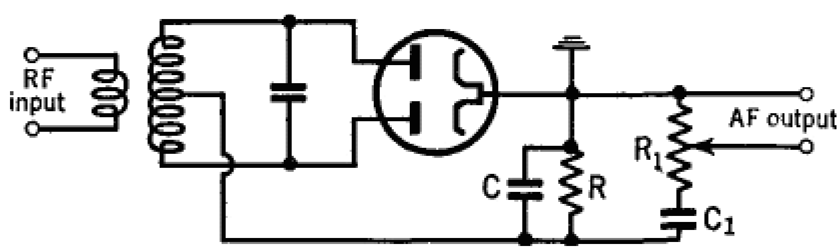


Fig. 7.3. A full-wave diode detector

In Fig. 17 D, full-wave rectification is used, the principle of operation being the same as described elsewhere. The audio-frequency voltages are taken out of the circuit through the condenser C_1 (= 0.1 μfd.) and the voltage divider or "volume control" R_1 (from 0.5 to 1 megohm).

CHAPTER – 8

IC 7806 – THE VOLTAGE REGULATOR

8.1 IC VOLTAGE REGULATORS

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. Although the internal construction of the IC is somewhat different from that described for discrete voltage regulator circuits, the external operation is much the same. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage. A power supply can be built using a transformer connected to the ac supply line to step the ac voltage to a desired amplitude, then rectifying that ac voltage, filtering with a capacitor and *RC* filter, if desired, and finally regulating the dc voltage using an IC regulator. The regulators can be selected for operation with load currents from hundreds of milliamperes to tens of amperes, corresponding to power ratings from milliwatts to tens of watts.

8.2 Three-Terminal Voltage Regulators

Figure 19.25 shows the basic connection of a three-terminal voltage regulator IC to a load. The fixed voltage regulator has an unregulated dc input voltage, *Vi*, applied to one input terminal, a regulated output dc voltage, *Vo*, from a second terminal, with the third terminal connected to ground. For a selected regulator, IC device specifications list a voltage range over which the input voltage can vary to maintain a regulated output voltage over a range of load current. The specifications also list the amount of output voltage change resulting from a change in load current (load regulation) or in input voltage (line regulation).

8.3 Fixed Positive Voltage Regulators

The series 78 regulators provide fixed regulated voltages from 5 to 24 V. Figure 19.26 shows how one such IC, a 7812, is connected to provide voltage regulation with output from this unit of _12 V dc. An unregulated input voltage *Vi* is filtered by capacitor *C*1 and connected to the IC's IN terminal. The IC's OUT terminal provides a regulated _12 V, which is filtered by capacitor *C*2 (mostly for any high frequency noise). The third IC terminal is connected to ground (GND). While the input voltage may vary over some permissible voltage range and the output load may vary over some acceptable range, the output voltage remains constant within specified voltage variation limits. These limitations are spelled out in the manufacturer's specification sheets. A table of positive voltage regulator ICs is provided in Table 19.1.

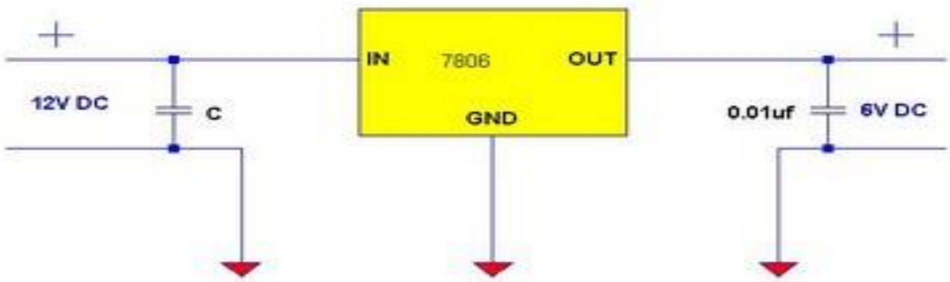


FIG 8.1**Positive Voltage Regulators in 7800 Series**

The connection of a 7812 in a complete voltage supply is shown in the connection of Fig. 19.27. The ac line voltage (120 V rms) is stepped down to 18 V rms across each half of the center-tapped transformer. A full-wave rectifier and capacitor filter then provides an unregulated dc voltage, shown as a dc voltage of about 22 V, with ac ripple of a few volts as input to the voltage regulator. The 7812 IC then provides an output that is a regulated _12 V dc.

8.4 POSITIVE VOLTAGE REGULATOR SPECIFICATIONS:

The specifications sheet of voltage regulators is typified by that shown in Fig. 19.28 for the group of series 7800 positive voltage regulators. Some consideration of a few of the more important parameters should be made.

8.4.1 OUTPUT VOLTAGE:

The specification for the 7812 shows that the output voltage is typically _12 V but could be as low as 11.5 V or as high as 12.5 V.

8.4.2 OUTPUT REGULATION:

The output voltage regulation is seen to be typically 4 mV, to a maximum of 100 mV (at output currents from 0.25 to 0.75 A). This information specifies that the output voltage can typically vary only 4 mV from the rated 12 V dc.

8.4.3 SHORT CIRCUIT OUTPUT CURRENT:

The amount of current is limited to typically 0.35 A if the output were to be short-circuited (presumably by accident or by another faulty component).

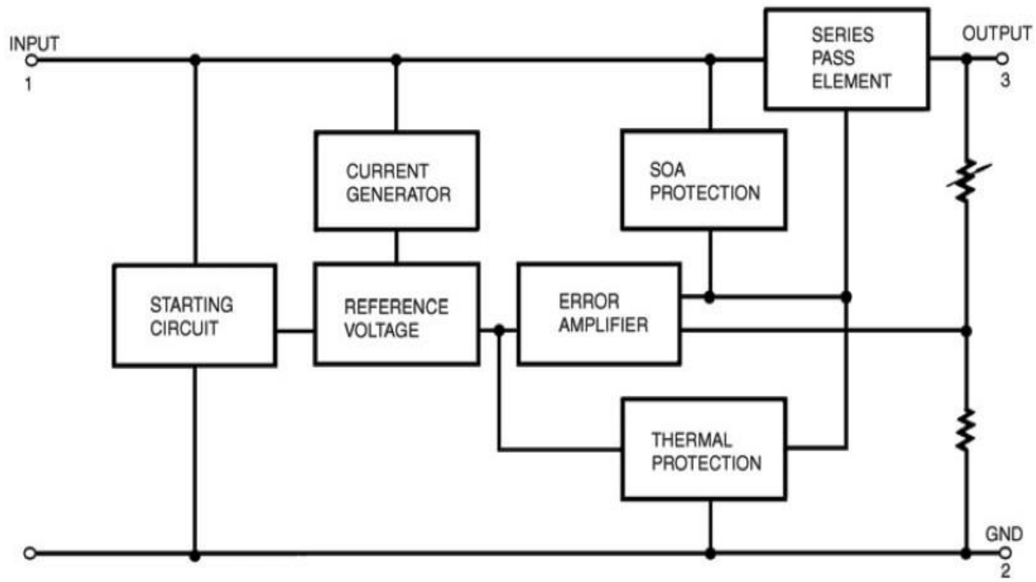
8.4.4 PEAK OUTPUT CURRENT:

While the rated maximum current is 1.5 A for this series of IC, the typical peak output current that might be drawn by a load is 2.2 A. This shows that although the manufacturer rates the IC as capable of providing 1.5 A, one could draw somewhat more current (possibly for a short period of time).

8.4.5 DROPOUT VOLTAGE:

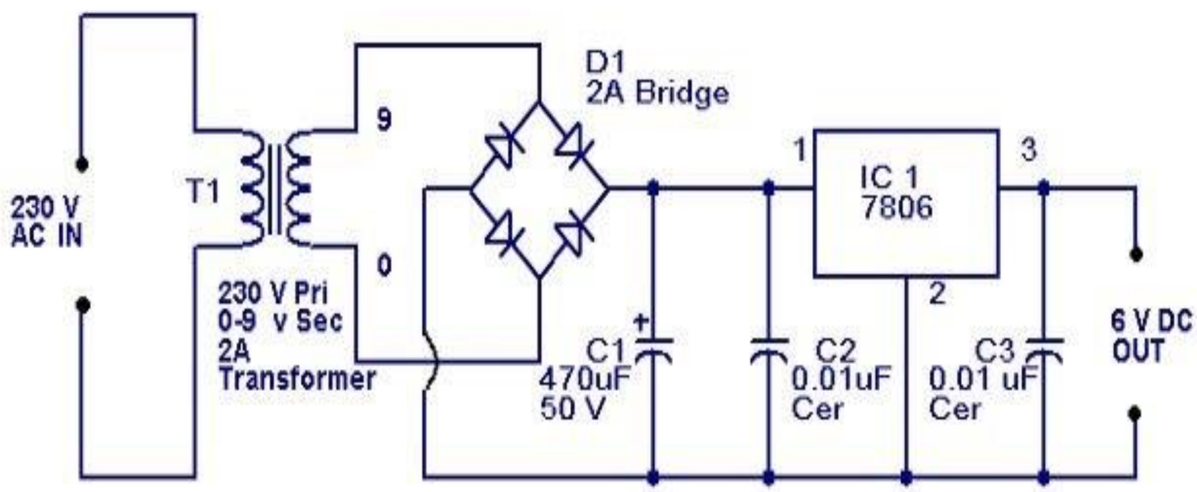
The dropout voltage, typically 2 V, is the minimum amount of voltage across the input–output terminals that must be maintained if the IC is to operate as a regulator. If the input voltage drops too low or the output rises so that at least 2 V is not maintained across the IC input–output, the IC will no longer provide voltage regulation. One therefore maintains an input voltage large enough to assure
that the dropout voltage is provided.

8.5 INTERNAL BLOCK DIAGRAM



{FIG 8.2 INTERNAL STRUCTURE}

8.6 VOLTAGE CONVERTER CIRCUIT USING IC 7806



{FIG 8.3 VOLTAGE CONVERTER CIRCUIT}

CHAPTER – 9

THE RELAY – A HANDLING TOOL

9.1 A BIT OVERVIEW

A relay is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism mechanically, but other operating principles are also used. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits, repeating the signal coming in from one circuit and re-transmitting it to another. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

A type of relay that can handle the high power required to directly control an electric motor is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

9.2 BASIC DESIGN AND OPERATION:

A simple electromagnetic relay consists of a coil of wire wrapped around a soft iron core, an iron yoke which provides a low reluctance path for magnetic flux, a movable iron armature, and one or more sets of contacts (there are two in the relay pictured). The armature is hinged to the yoke and mechanically linked to one or more sets of moving contacts. It is held in place by a spring so that when the relay is de-energized there is an air gap in the magnetic circuit. In this condition, one of the two sets of contacts in the relay pictured is closed, and the other set is open. Other relays may have more or fewer sets of contacts depending on their function. The relay in the picture also has a wire connecting the armature to the yoke. This ensures continuity of the circuit between the moving contacts on the armature, and the circuit track on the printed circuit board (PCB) via the yoke, which is soldered to the PCB.

When an electric current is passed through the coil it generates a magnetic field that activates the armature, and the consequent movement of the movable contact(s) either makes or breaks (depending upon construction) a connection with a fixed contact. If the set of contacts was closed when the relay was de-energized, then the movement opens the contacts and breaks the connection, and vice versa if the contacts were open. When the current to the coil is switched off, the armature is returned by a force, approximately half as strong as the magnetic force, to its relaxed position. Usually this force is provided by a spring, but gravity is also used commonly in industrial motor starters. Most relays are manufactured to operate quickly. In a low-voltage application this reduces noise; in a high voltage or current application it reduces arcing.

When the coil is energized with direct current, a diode is often placed across the coil to dissipate the energy from the collapsing magnetic field at deactivation, which would otherwise generate a voltage spike dangerous to semiconductor circuit components. Some automotive relays include a diode inside the relay case. Alternatively, a contact protection network consisting of a capacitor and resistor in series (snubber circuit) may absorb the surge. If the coil is designed to be energized with alternating current (AC), a small copper "shading ring" can be crimped to the end of the solenoid, creating a small out-of-phase current which increases the minimum pull on the armature during the AC cycle.[1]

A solid-state relay uses a thyristor or other solid-state switching device, activated by the control signal, to switch the controlled load, instead of a solenoid. An optocoupler (a light-emitting diode (LED) coupled with a photo transistor) can be used to isolate control and controlled circuits.

9.3TYPES

9.3.1 LATCHING RELAY

A *latching relay* has two relaxed states (bistable). These are also called "impulse", "keep", or "stay" relays. When the current is switched off, the relay remains in its last state. This is achieved with a solenoid operating a ratchet and cam mechanism, or by having two opposing coils with an over-center spring or permanent magnet to hold the armature and contacts in position while the coil is relaxed, or with a remanent core. In the ratchet and cam example, the first pulse to the coil turns the relay on and the second pulse turns it off. In the two coil example, a pulse to one coil turns the relay on and a pulse to the opposite coil turns the relay off. This type of relay has the advantage that one coil consumes power only for an instant, while it is being switched, and the relay contacts retain this setting across a power outage. A remanent core latching relay requires a current pulse of opposite polarity to make it change state.

9.3.2 REED REALY

A reed relay is a reed switch enclosed in a solenoid. The switch has a set of contacts inside an evacuated or inert gas-filled glass tube which protects the contacts against atmospheric corrosion; the contacts are made of magnetic material that makes them move under the influence of the field of the enclosing solenoid. Reed relays can switch faster than larger relays, require only little power from the control circuit, but have low switching current and voltage ratings. In addition, the reeds can become magnetized over time, which makes them stick 'on' even when no current is present; changing the orientation of the reeds with respect to the solenoid's magnetic field will fix the problem.

9.3.3 MERCURY-WETTED RELAY

A mercury-wetted reed relay is a form of reed relay in which the contacts are wetted with mercury. Such relays are used to switch low-voltage signals (one volt or less) where the mercury reduces the contact resistance and associated voltage drop, for low-current signals where surface contamination may make for a poor contact, or for high-speed applications where the mercury eliminates contact bounce. Mercury wetted relays are position-sensitive and must be mounted vertically to work properly. Because of the toxicity and expense of liquid mercury, these relays are now rarely used. See also mercury switch.

9.3.4 POLARIZED RELAY

A polarized relay placed the armature between the poles of a permanent magnet to increase sensitivity. Polarized relays were used in middle 20th Century telephone exchanges to detect faint pulses and correct telegraphic distortion. The poles were on screws, so a technician could first adjust them for maximum sensitivity and then apply a bias spring to set the critical current that would operate the relay.

9.3.5 MACHINE TOOL RELAY

A machine tool relay is a type standardized for industrial control of machine tools, transfer machines, and other sequential control. They are characterized by a large number of contacts (sometimes extendable in the field) which are easily converted from normally-open to normally-closed status, easily replaceable coils, and a form factor that allows compactly installing many relays in a control panel. Although such relays once were the backbone of automation in such industries as automobile assembly, the programmable logic controller (PLC) mostly displaced the machine tool relay from sequential control applications.

A relay allows circuits to be switched by electrical equipment: for example, a timer circuit with a relay could switch power at a preset time. For many years relays were the standard method of controlling industrial electronic systems. A number of relays could be used together to carry out complex functions (relay logic). The principle of relay logic is based on relays which energize and de-energize associated contacts. Relay logic is the predecessor of ladder logic, which is commonly used in Programmable logic controllers.

9.3.6 RATCHET RELAY

This is again a clapper type relay which does not need continuous current through its coil to retain its operation.

9.3.7 CONTACTOR RELAY

A contactor is a very heavy-duty relay used for switching electric motors and lighting loads, although contactors are not generally called relays. Continuous current ratings for common contactors range from 10 amps to several hundred amps. High-current contacts are made with alloys containing silver. The unavoidable arcing causes the contacts to oxidize; however, silver oxide is still a good conductor.[2] Such devices are often used for motor starters. A motor starter is a contactor with overload protection devices attached. The overload sensing devices are a form of heat operated relay where a coil heats a bi-metal strip, or where a solder pot melts, releasing a spring to operate auxiliary contacts. These auxiliary contacts are in series with the coil. If the overload senses excess current in the load, the coil is de-energized. Contactor relays can be extremely loud to operate, making them unfit for use where noise is a chief concern.

9.3.8 SOLID- STATE RELAY

A solid state relay (SSR) is a solid state electronic component that provides a similar function to an electromechanical relay but does not have any moving components, increasing long-term reliability. With early SSR's, the tradeoff came from the fact that every transistor has a small voltage drop across it. This voltage drop limited the amount of current a given SSR could handle. The minimum voltage drop for such a relay is equal to the voltage drop across one transistor (~0.6-2.0 volts), and is a function of the material used to make the transistor (typically silicon). As transistors improved, higher current SSR's, able to handle 100 to 1,200 Amperes, have become commercially available. Compared to electromagnetic relays, they may be falsely triggered by transients.

9.3.9 SOLID STATE CONTACTOR RELAY

A solid state contactor is a heavy-duty solid state relay, including the necessary heat sink, used for switching electric heaters, small electric motors and lighting loads; where frequent on/off cycles are required. There are no moving parts to wear out and there is no contact bounce due to vibration. They are activated by AC control signals or DC control signals from Programmable logic controller (PLCs), PCs, Transistor-transistor logic (TTL) sources, or other microprocessor and microcontroller controls.

9.3.10 BUCHHOLZ RELAY

A Buchholz relay is a safety device sensing the accumulation of gas in large oil-filled transformers, which will alarm on slow accumulation of gas or shut down the transformer if gas is produced rapidly in the transformer oil.

9.3.11 FORCED-GUIDED CONTACTS RELAY

A forced-guided contacts relay has relay contacts that are mechanically linked together, so that when the relay coil is energized or de-energized, all of the linked contacts move together. If one set of contacts in the relay becomes immobilized, no other contact of the same relay will be able to move. The function of forced-guided contacts is to enable the safety circuit to check the status of the relay. Forced-guided contacts are also known as "positive-guided contacts", "captive contacts", "locked contacts", or "safety relays".

9.3.12 OVERLOAD PROTECTION RELAY

Electric motors need overcurrent protection to prevent damage from over-loading the motor, or to protect against short circuits in connecting cables or internal faults in the motor windings.[3] One type of electric motor overload protection relay is operated by a heating element in series with the electric motor. The heat generated by the motor current heats a bimetallic strip or melts solder, releasing a spring to operate contacts. Where the overload relay is exposed to the same environment as the motor, a useful though crude compensation for motor ambient temperature is provided.

9.4 POLE AND THROW

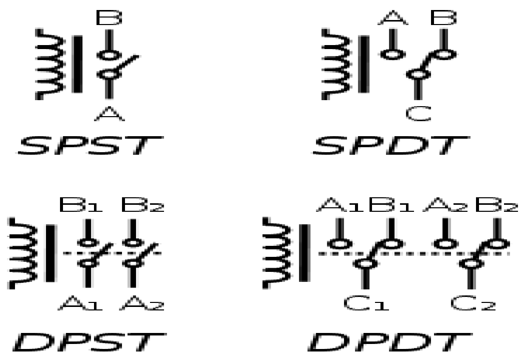


FIG 9.1 Circuit symbols of relays. (C denotes the common terminal in SPDT and DPDT types.)

Since relays are switches, the terminology applied to switches is also applied to relays. A relay will switch one or more *poles*, each of whose contacts can be *thrown* by energizing the coil in one of three ways:

- Normally-open (**NO**) contacts connect the circuit when the relay is activated; the circuit is disconnected when the relay is inactive. It is also called a **Form A** contact or "make" contact. **NO** contacts can also be distinguished as "early-make" or **NOEM**, which means that the contacts will close before the button or switch is fully engaged.
- Normally-closed (**NC**) contacts disconnect the circuit when the relay is activated; the circuit is connected when the relay is inactive. It is also called a **Form B** contact or "break" contact. **NC** contacts can also be distinguished as "late-break" or **NCLB**, which means that the contacts will stay closed until the button or switch is fully disengaged.
- Change-over (**CO**), or double-throw (**DT**), contacts control two circuits: one normally-open contact and one normally-closed contact with a common terminal. It is also called a **Form C** contact or "transfer" contact ("break before make"). If this type of contact utilizes a "make before break" functionality, then it is called a **Form D** contact.

The following designations are commonly encountered:

- **SPST** – Single Pole Single Throw. These have two terminals which can be connected or disconnected. Including two for the coil, such a relay has four terminals in total. It is ambiguous whether the pole is normally open or normally closed. The terminology "SPNO" and "SPNC" is sometimes used to resolve the ambiguity.
- **SPDT** – Single Pole Double Throw. A common terminal connects to either of two others. Including two for the coil, such a relay has five terminals in total.
- **DPST** – Double Pole Single Throw. These have two pairs of terminals. Equivalent to two SPST switches or relays actuated by a single coil. Including two for the coil, such a relay has six terminals in total. The poles may be Form A or Form B (or one of each).
- **DPDT** – Double Pole Double Throw. These have two rows of change-over terminals. Equivalent to two SPDT switches or relays actuated by a single coil. Such a relay has eight terminals, including the coil.

The "S" or "D" may be replaced with a number, indicating multiple switches connected to a single actuator. For example 4PDT indicates a four pole double throw relay (with 14 terminals).

9.5 APPLICATIONS

Relays are used to and for:

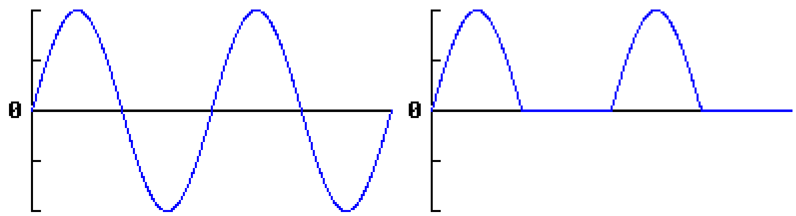
- Amplify a digital signal, switching a large amount of power with a small operating power. Some special cases are:
    - A telegraph relay, repeating a weak signal received at the end of a long wire
    - Controlling a high-voltage circuit with a low-voltage signal, as in some types of modems or audio amplifiers,
    - Controlling a high-current circuit with a low-current signal, as in the starter solenoid of an automobile,
- Detect and isolate faults on transmission and distribution lines by opening and closing circuit breakers (protection relays),

- Isolate the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch. The latter is often applied to control office lighting as the low voltage wires are easily installed in partitions, which may be often moved as needs change. They may also be controlled by room occupancy detectors to conserve energy,
- Logic functions. For example, the boolean AND function is realised by connecting normally open relay contacts in series, the OR function by connecting normally open contacts in parallel. The change-over or Form C contacts perform the XOR (exclusive or) function. Similar functions for NAND and NOR are accomplished using normally closed contacts. The Ladder programming language is often used for designing relay logic networks.
    - The application of Boolean Algebra to relay circuit design was formalized by Claude Shannon in A Symbolic Analysis of Relay and Switching Circuits
    - Early computing. Before vacuum tubes and transistors, relays were used as logical elements in digital computers. See electro-mechanical computers such as ARRA (computer), Harvard Mark II, Zuse Z2, and Zuse Z3.
    - Safety-critical logic. Because relays are much more resistant than semiconductors to nuclear radiation, they are widely used in safety-critical logic, such as the control panels of radioactive waste-handling machinery.
- Time delay functions. Relays can be modified to delay opening or delay closing a set of contacts. A very short (a fraction of a second) delay would use a copper disk between the armature and moving blade assembly. Current flowing in the disk maintains magnetic field for a short time, lengthening release time. For a slightly longer (up to a minute) delay, a dashpot is used. A dashpot is a piston filled with fluid that is allowed to escape slowly. The time period can be varied by increasing or decreasing the flow rate. For longer time periods, a mechanical clockwork timer is installed.
- Vehicle battery isolation. A 12v relay is often used to isolate any second battery in cars, 4WDs, RVs and boats.
- Switching to a standby power supply.

CHAPTER – 10

THE RECTIFIER – A DIODE APPLICATION

10.1 BRIEF INTRODUCTION

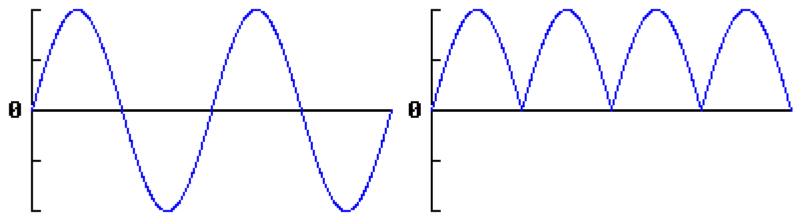


{FIG 10.1 SINUSOIDAL INPUT AND HALF WAVE OUTPUT}

As we have noted when looking at the Elements of a Power Supply, the purpose of the rectifier section is to convert the incoming ac from a transformer or other ac power source to some form of pulsating dc. That is, it takes current that flows alternately in both directions as shown in the first figure to the right, and modifies it so that the output current flows only in one direction, as shown in the second and third figures below.

The circuit required to do this may be nothing more than a single diode, or it may be considerably more complex. However, all rectifier circuits may be classified into one of two categories, as follows:

- Half-Wave Rectifiers. An easy way to convert ac to pulsating dc is to simply allow half of the ac cycle to pass, while blocking current to prevent it from flowing during the other half cycle. The figure to the right shows the resulting output. Such circuits are known as *half-wave rectifiers* because they only work on half of the incoming ac wave.

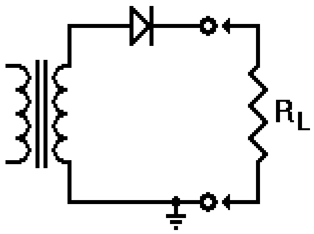


{FIG 10.2 SINUSOIDAL INPUT AND CORRESPONDING FULL WAVE OUTPUT}

- Full-Wave Rectifiers. The more common approach is to manipulate the incoming ac wave so that both halves are used to cause output current to flow in the same direction. The resulting waveform is shown to the right. Because these circuits operate on the entire incoming ac wave, they are known as *full-wave rectifiers*.

Rectifier circuits may also be further clasified according to their configuration, as we will see below.

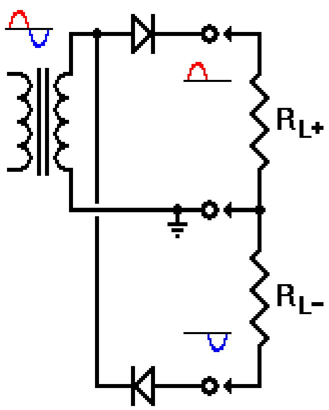
10.2 THE HALF WAVE RECTIFIER



{FIG 10.3 BASIC HALF WAVE RECTIFIER}

The simplest rectifier circuit is nothing more than a diode connected in series with the ac input, as shown to the right. Since a diode passes current in only one direction, only half of the incoming ac wave will reach the rectifier output. Thus, this is a basic half-wave rectifier.

The orientation of the diode matters; as shown, it passes only the positive half-cycle of the ac input, so the output voltage contains a positive dc component. If the diode were to be reversed, the negative half-cycle would be passed instead, and the dc component of the output would have a negative polarity. In either case, the DC component of the output waveform is $v_p/\pi = 0.3183v_p$, where v_p is the peak voltage output from the transformer secondary winding.

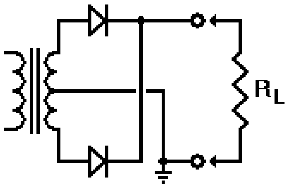


{FIG 10.4 USE OF HWR IN CIRCUIT}

It is also quite possible to use two half-wave rectifiers together, as shown in the second figure to the right. This arrangement provides both positive and negative output voltages, with each output utilizing half of the incoming ac cycle.

Note that in all cases, the lower transformer connection also serves as the common reference point for the output. It is typically connected to the common ground of the overall circuit. This can be very important in some applications. The transformer windings are of course electrically insulated from the iron core, and that core is normally grounded by the fact that it is bolted physically to the metal chassis (box) that supports the entire circuit. By also grounding one end of the secondary winding, we help ensure that this winding will never experience even momentary voltages that might overload the insulation and damage the transformer.

10.3 THE FULL WAVE RECTIFIER

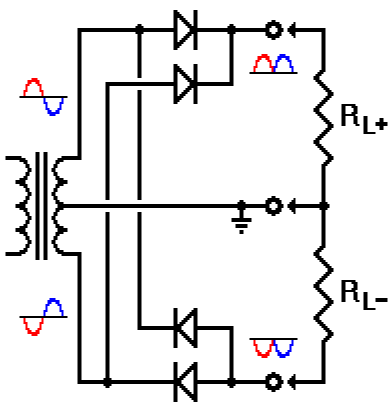


{FIG 10.5 BASIC FULL WAVE RECTIFIER}

While the half-wave rectifier is very simple and does work, it isn't very efficient. It only uses half of the incoming ac cycle, and wastes all of the energy available in the other half. For greater efficiency, we would like to be able to utilize both halves of the incoming ac. One way to accomplish this is to double the size of the secondary winding and provide a connection to its center. Then we can use two separate half-wave rectifiers on alternate half-cycles, to provide full-wave rectification. The circuit is shown to the right.

Because both half-cycles are being used, the DC component of the output waveform is now $2v_p/\pi = 0.6366v_p$, where v_p is the peak voltage output from *half* the transformer secondary winding, because only half is being used at a time.

This rectifier configuration, like the half-wave rectifier, calls for one of the transformer's secondary leads to be grounded. In this case, however, it is the center connection, generally known as the *center tap* on the secondary winding.



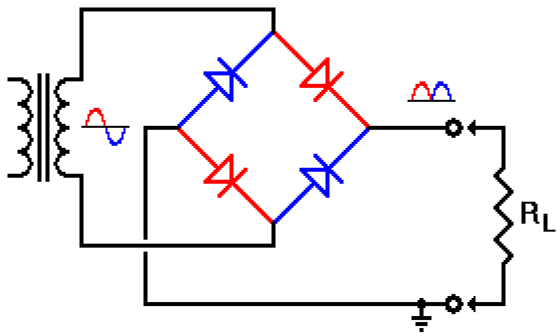
{FIG 10.6 USE OF FWR IN CIRCUIT}

The full-wave rectifier can still be configured for a negative output voltage, rather than positive. In addition, as shown to the right, it is quite possible to use two full-wave rectifiers to get outputs of both polarities at the same time.

The full-wave rectifier passes both halves of the ac cycle to either a positive or negative output. This makes more energy available to the output, without large intervals when no energy is provided at all. Therefore, the full-wave rectifier is more efficient than the half-wave rectifier. At the same time, however, a full-wave rectifier providing only a single output polarity does require a secondary winding that is twice as big as the half-wave rectifier's secondary, because only half of the secondary winding is providing power on any one half-cycle of the incoming ac.

Actually, it isn't all that bad, because the use of both half-cycles means that the current drain on the transformer winding need not be as heavy. With power being provided on both half-cycles, one half-cycle doesn't have to provide enough power to carry the load past an unused half-cycle. Nevertheless, there are some occasions when we would like to be able to use the entire transformer winding at all times, and still get full-wave rectification with a single output polarity.

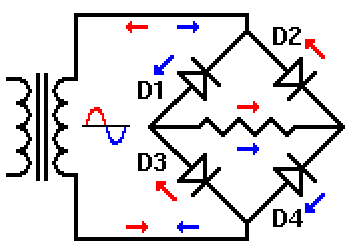
10.4 THE FULL-WAVE BRIDGE RECTIFIER



{FIG 10.7 BASIC BRIDGE RECTIFIER}

The four-diode rectifier circuit shown to the right serves very nicely to provide full-wave rectification of the ac output of a single transformer winding. The diamond configuration of the four diodes is the same as the resistor configuration in a Wheatstone Bridge. In fact, any set of components in this configuration is identified as some sort of bridge, and this rectifier circuit is similarly known as a *bridge rectifier*.

If you compare this circuit with the dual-polarity full-wave rectifier above, you'll find that the connections to the diodes are the same. The only change is that we have removed the center tap on the secondary winding, and used the negative output as our ground reference instead. This means that the transformer secondary is never directly grounded, but one end or the other will always be close to ground, through a forward-biased diode. This is not usually a problem in modern circuits.



{FIG 10.8 ORIENTATION OF BRIDGE RECTIFIER}

To understand how the bridge rectifier can pass current to a load in only one direction, consider the figure to the right. Here we have placed a simple resistor as the load, and we have numbered the four diodes so we can identify them individually.

During the positive half-cycle, shown in red, the top end of the transformer winding is positive with respect to the bottom half. Therefore, the transformer pushes electrons from its bottom end, through D3 which is forward biased, and through the load resistor in the direction shown by the red arrows. Electrons then continue through the forward-biased D2, and from there to the top of the transformer winding. This forms a complete circuit, so current can indeed flow. At the same time, D1 and D4 are reverse biased, so they do not conduct any current.

During the negative half-cycle, the top end of the transformer winding is negative. Now, D1 and D4 are forward biased, and D2 and D3 are reverse biased. Therefore, electrons move through D1, the resistor, and D4 in the direction shown by the blue arrows. As with the positive half-cycle, electrons move through the resistor from left to right.

In this manner, the diodes keep switching the transformer connections to the resistor so that current always flows in only one direction through the resistor. We can replace the resistor with any other circuit, including more power supply circuitry (such as the filter), and still see the same behavior from the bridge rectifier.

CHAPTER – 11

SOURCE CODE – THE PERFORMER

```
#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#include <avr/interrupt.h>


#define BAUD 4800

#define UBRR_VAL F_CPU/16/BAUD-1


#define DEBUG_MODE


#define BURST_TIME      70

#define READY_TIME      150

#define TIME_OUT        300

#define MAX             1020

#define MIN             10



volatile char ready, burst, first;

int timer;



void start_timer(void);

void stop_timer(void);
```

```
void init_uart(unsigned int ubrr) {
	/* Set baud rate */
	UBRRH = (unsigned char)(ubrr>>8 ) ;
	UBRRL = (unsigned char)(ubrr);
	/* Enable receiver and transmitter */
	UCSRB |= (1<<RXEN);
	UCSRB |= (1<<TXEN);
	/* Set frame format: 8data, 2stop bit */
	UCSRC = (1<<URSEL)|(1<<USBS)|(3<<UCSZ0);
}

void send_char(unsigned char data) {
	/* Wait for empty transmit buffer */
	while (!(UCSRA & (1<<UDRE)));
		/* Put data into buffer, sends the data */
		UDR = data;
}

void send_string(char *data) {
	while (*data) {
		send_char(*data);
		data++;
	}
}

void adc_init()
{
```

```
		ADMUX = 0x60; //avcc with external cap, result left adjust
		ADCSRA = 0x85; //enable adc, prescaler div 32 (4MHz / 32 = 125kHz)
}

void timer_init()
{
		TIFR |= (1<<TOV0);			//set interrupt on overflow
		TIMSK |= (1<<TOIE0);
		TCNT0 = 223;
}

ISR (TIMER0_OVF_vect)
{
		TCNT0 = 223;
		timer++;
		if (timer == BURST_TIME)
			{
				#ifdef DEBUG_MODE
				send_string("Burst\n\r");
				#endif
				burst = 1;
				ready = 0;
			} else
		if (timer == READY_TIME)
			{
				burst = 0;
				ready = 1;
```

```
                #ifdef DEBUG_MODE
                send_string("Ready\n\r");
                #endif
            } else
        if (timer == TIME_OUT)
            {
                #ifdef DEBUG_MODE
                send_string("Time out\n\r");
                #endif
                timer = 0;
                burst = 0;
                ready = 0;
                first = 0;
                stop_timer();
            }
}

void start_timer()
{
        TCCR0 |= (1<<CS02);
}

void stop_timer()
{
        TCCR0 &= ~(1<<CS02);
}
```

```
int ADC_convert(int channel)
{
        ADMUX |= (0x0F & (channel));
        ADCSRA |= (1<<ADSC);
        return ADCH;
}

int main()
{
        int value, i, dummy;
        char buffer[1];

        adc_init();
        init_uart(UBRR_VAL);
        timer_init();
        send_string("Clapper V1.2 \n\r");

        i = 100;
        ready = 0;
        burst = 0;
        first = 0;

        DDRB = 0x04;
        PORTB &= ~(1<<PB2);
        PORTB |= (1<<PB0) | (1<<PB1);

        sei();
```

```
while (1)
{
        if (bit_is_clear(PINB,PB0))
                {
                        _delay_ms(250);
                        #ifdef DEBUG_MODE
                        send_string("Sensibility up\n\r");
                        #endif
                        if (i < MAX)
                        {
                                i +=10;
                        } else
                                i = MAX;
                        #ifdef DEBUG_MODE
                        send_string(itoa(i,buffer,10));
                        send_string("\n\r");
                        #endif
                }
        if (bit_is_clear(PINB,PB1))
                {
                        _delay_ms(250);
                        #ifdef DEBUG_MODE
                        send_string("Sensibility down\n\r");
                        #endif
                        if (i > MIN)
                        {
```

```
			i -=10;
		} else
			i = MIN;
		#ifdef DEBUG_MODE
		send_string(itoa(i,buffer,10));
		send_string("\n\r");
		#endif
	}

//	value = (ADCH * 5) / 255; get the voltage on adc0
	value = ADC_convert(0x00);
	if (value >= i)
		{
		#ifdef DEBUG_MODE
		send_string("Beat\n\r");
		#endif

		if (!first)
			{
			start_timer();
			first = 1;

			#ifdef DEBUG_MODE
			send_string("First Pulse\n\r");
			#endif
		} else
```

```
if (burst)
        {
                timer = 0;
                first = 0;
                burst = 0;
                stop_timer();

                #ifdef DEBUG_MODE
                send_string("False Pulse\n\r");
                #endif
        } else {

if (ready)
        {
                stop_timer();
                timer = 0;
                first = 0;
                ready = 0;

                PORTB ^= (1<<PB2);

                #ifdef DEBUG_MODE
                send_string("ON/OFF Lights\n\r");
                #endif

                value = 0;
                ADCH = 0;
```

```
                        _delay_ms(10);

                        value = ADC_convert(0x00);

                        _delay_ms(1000);


                    }

}

                        _delay_ms(40);



                }

        }

}
```