AN 8080 BASED MICROCOMPUTER

by

-Objective: To build a quality microcomputer cheaply and easily.

Building the Hardware

A computer has four basic parts: a central processing unit (CPU), a data storage system (memory), some kind of input/output device, and a data bus (backplane) to connect the above three together. To build my computer, I followed this procedure:

1) Define my objectives.
   -Decide what I want the computer to do.
   -Decide how much money and work I want to put into it.

2) Design my computer, keeping the above in mind.
   -Choose or design a bus structure.
   -Choose or design a CPU.
   -Choose or design a memory system.
   -Design an input/output system.

3) Buy the parts for the computer, build it, and debug it.

My computer has the following features:

- S-100 bus structure, with 12 slots.
- 8080A based CPU.
- 2K bytes of static memory.
- A front panel that is completely independant of the CPU.

In the future, I hope to add more memory and better input/output.

Software Development

After I finished building the computer's basic hardware, I wrote some software. This software includes a time delay program, an addition-subtraction program, and an 8-bit multiply program. In the future, I hope to write more software, including a floating-point math routine and a high-level language.

P R O G R E S S R E P O R T

Month, year	Progress
Sep 76 -	I was introduced to the school's computer. I became interested in programming.
Oct 76 -	I became a more experienced programmer. Other people also became interested in the school computer; they used it mostly for playing games (like Startrek). The computer started malfunctioning often. I decided that I wanted to own a computer.
Nov 76 -	I got books on computers from various sources. Unfortunately, these books were out of date, although I didn't know this at the time. Using these books, I learned the value of Boolean algebra in logic design. Using my knowledge of Boolean algebra, I designed and built a base 8 to LED display decoder.
Dec 76 -	I started working on a computer program to simplify Boolean expressions (thus simplifying logic design).
-Jan 77 -	I did more work on this program, and began debugging it. I bought a CMOS data book.
Feb 77 -	I perfected the program and prepared to enter it in the science fair.
Mar 77 -	City and Regional science fairs. I finished a 16-digit LED display. I ordered 8 2102 RAM chips. I subscribed to Kilobaud magazine.
Apr 77 -	I bought an 8080A chip and began designing a bus for it. I chose a 44 pin bus.
May 77 -	I worked more on my 16 digit output display. I started building a 1K * 8 bit memory board for my bus.
Jun 77 -	I built a power supply for my computer. I discarded my old memory board and started on a new one.
Jul 77 -	I finished my CPU board and tested it. It worked! However, I had trouble building a comatible memory board. Because of this, I chose the S-100 bus. (After some looking around.)
Aug 77 -	Vacationing.
Sep 77 -	I ordered an S-100 backplane and an S-100 compatible 8K memory board (without parts). I began working on a front panel.

PROGRESS REPORT

Month, year	Progress
Oct 77 -	I ordered parts for the memory board. I had some problems getting the parts. I worked on my front panel.
Nov 77 -	I tried to make an S-100 adapter for my CPU. It was a disastor, so I discarded my 44-pin CPU. At school, I began a program to convert 8080 assembly code into machine code.
Dec 77 -	I ordered a Z-80 manual (which still has not arrived). I finished the 8080 coding program and wrote some programs for the 8080.
Jan 78 -	Because the Z-80 manual had not yet arrived, I ordered an 8080A CPU board (without parts). After receiving the CPU board, I ordered parts for the CPU.
Feb 78 -	The parts for the CPU arrived. I put it together, and debugged it (due to a documentation error). I discovered that my memory board was not completely compatible with my CPU board. I ordered the parts for a new front panel board.
Mar 78 -	The parts for the front panel board arrived.

P R E S E N T S T A T U S
A N D
F U T U R E P L A N S

This computer system is based on the Wameco, Inc. CPU board. This board uses the 8080A microprocessor chip and is S-100 compatible. The memory board is the Wameco, Inc. 8K static memory board. Only 2K of memory is installed.

In the future, I hope to add extra memory (RAM and ROM), and hook up a tape recorder interface for mass storage. For input/output, I hope to add an ASCII keyboard, and a CRT.

I will also need software. I want a tape loading routine, a floating point arithmetic routine, an assembler, and possibly even a high level language interpreter.

Present Description

Catagory	Description
-CPU	
-Microprocessor chip	8080A
-Byte (word) length	8 bits
-Clock frequency	2 MHz
-Board manufacturer	Wameco, Inc.
-Memory (static RAM)	
-Board capacity	
-Maximum possible	8K * 8
-Present	2K * 8
-Chip used	2102
-Access time	500 nsec
-Board manufacturer	Wameco, Inc.
-Backplane	
-Bus	S-100
-Slots, number of	12
-Board manufacturer	Wameco, Inc.
-Input/output	front panel

F U T U R E P L A N S

Catagory	Order of completion	Description
-Memory		
-RAM (static)	7	8K or more
-ROM	?	??
-Mass storage	2	Tape recorder interface
-Input/output	1	Front panel
	4	Keyboard
	5	CRT display
-Software	3	Tape loader
	6	Assembler
	8	Arithmetic (floating point)
	9	High level language interpreter

C H O O S I N G A B U S S T R U C T U R E

I chose the S-100 bus structure because of the variety, availability, and low cost of S-100 peripherals. Most other bus structures have a few expensive peripherals that are hard to find.

C H O O S I N G A P R O C E S S O R

The four most popular microprocessors are: the 8080, the Z-80, the 6502, and the 6800. I chose the 8080 because of its WAIT states, its 6 general purpose registers, its S-100 compatibility, and its low cost.

Note : At the time I chose the 8080, the Z-80 chip costed $ 80.

BUSSES AND BACKPLANES

Characteristic	S-100	Most others
-Number of pins	100	40-50
-Cost/slot		
-Edge connector	$ 4	$ 2 - $ 3
-Backplane	$ 3	$ 1 - $ 3
-Total	$ 7	$ 3 - $ 6
-Peripherals		
-Number of vendors	Many	Few
-Variety	Excellent. Everything from the "Better Bug Trap" to the "Computalker."	Poor. Only memory, CPU, I/O.
-Cost		
-CPU		
-8080	$ 90	not available separately
-Z-80	$ 140	not available separately
-6502	incompatible	$ 55
-6800	incompatible	$ 80
-Memory, static		
-4K, 2102 based	$ 70 - $ 90	$ 70 - $ 150
-8K, 2102 based	$ 120 - $ 150	$ 150 - $ 300
-8K, other		$ 200 - $ 300
-16K, other	$ 300 - $ 400	
-32K, other	$ 800 - $1000	
-Memory, dynamic		
-8K	$ 110 - $ 150	$ 150 - $ 300
-16K	$ 200 - $ 300	
-32K	$ 450 - $ 550	
-ROM		
-4K	$ 60 - $ 70	
-8K		
-16K	$ 60 - $ 90	
-Other	Available at low prices.	Not available or hard to find. High prices.

CPU INSTRUCTION SETS

Characteristic	8080	Z-80	6502	6800
-Number of instructions	66	158	56	72
-Arithmetic				
-Binary	Add, subtract	Add, subtract	Add, subtract	Add, subtract
-Decimal	Add	Add, subtract	Add, subtract	Add
-Logical				
-And	Yes	Yes	Yes	Yes
-Or	Yes	Yes	Yes	Yes
-Exclusive or	Yes	Yes	Yes	Yes
-Invert	Yes	Yes	No	Yes
-Branch				
-Conditional	Yes	Yes	Yes	Yes
-to subroutine	Yes	Yes	Yes	Yes
-Registers				
-Accumulator	1	2	1	2
-General	6	12	0	0
-Index	0	2, 16 bit	2, 8 bit	1, 16 bit
-Data pointer	3	6	0	0
-Stack pointer	16 bit	16 bit	8 bit	16 bit
-Program counter	16 bit	16 bit	16 bit	16 bit
-Addressing modes				
-Bit level	No	Yes	No	No
-Direct	Yes	Yes	Yes	Yes
-Immediate	Yes	Yes	Yes	Yes
-Implied	Yes	Yes	Yes	Yes
-Indexed	No	8 bit offset	16 bit offset	8 bit offset
-Register indirect	Yes	Yes	No	No
-Relative	No	8 bit offset	8 bit offset	8 bit offset

C P U H A R D W A R E

Charactaristic	8080	Z-80	6502	6800
-Control functions				
-Vectored interrupts				
-Reset	Yes	Yes	Yes	Yes
-Multiple priority	8	8	2	2
-Software	Yes	Yes	Yes	Yes
-Wait				
-on read	Yes	Yes	Yes	No
-on write	Yes	Yes	No	No
-Execution speed				
-Clock frequency (max)	2 MHz	4 MHz	2 MHz	1 MHz
-Instruction cycle	2000 nsec	1000 nsec	1000 nsec	2000 nsec
-Hardware and support				
-Voltage levels				
- +12 volt	Yes	No	No	No
- + 5 volt	Yes	Yes	Yes	Yes
- - 5 volt	Yes	No	No	No
-Power consumption	.8 W	---	---	.6 W
-Cost of chip	$ 10	$ 30	$ 15	$ 18
-Cost of support	$ 80	$ 110	$ 40	$ 60
-Total CPU cost	$ 90	$ 140	$ 55	$ 78
-Availability of				
-Chip	Good	Good	Fair	Fair
-Support	Good	Fair	Good	Good
-CPU kits (S-100)	Good	Good	Poor	Poor
-Software	Good	Good	Fair	Fair
-Documentation	Good		Fair	Good
-Manufacturer's reputation	Good	---	Fair	Excellent

T H E F R O N T P A N E L

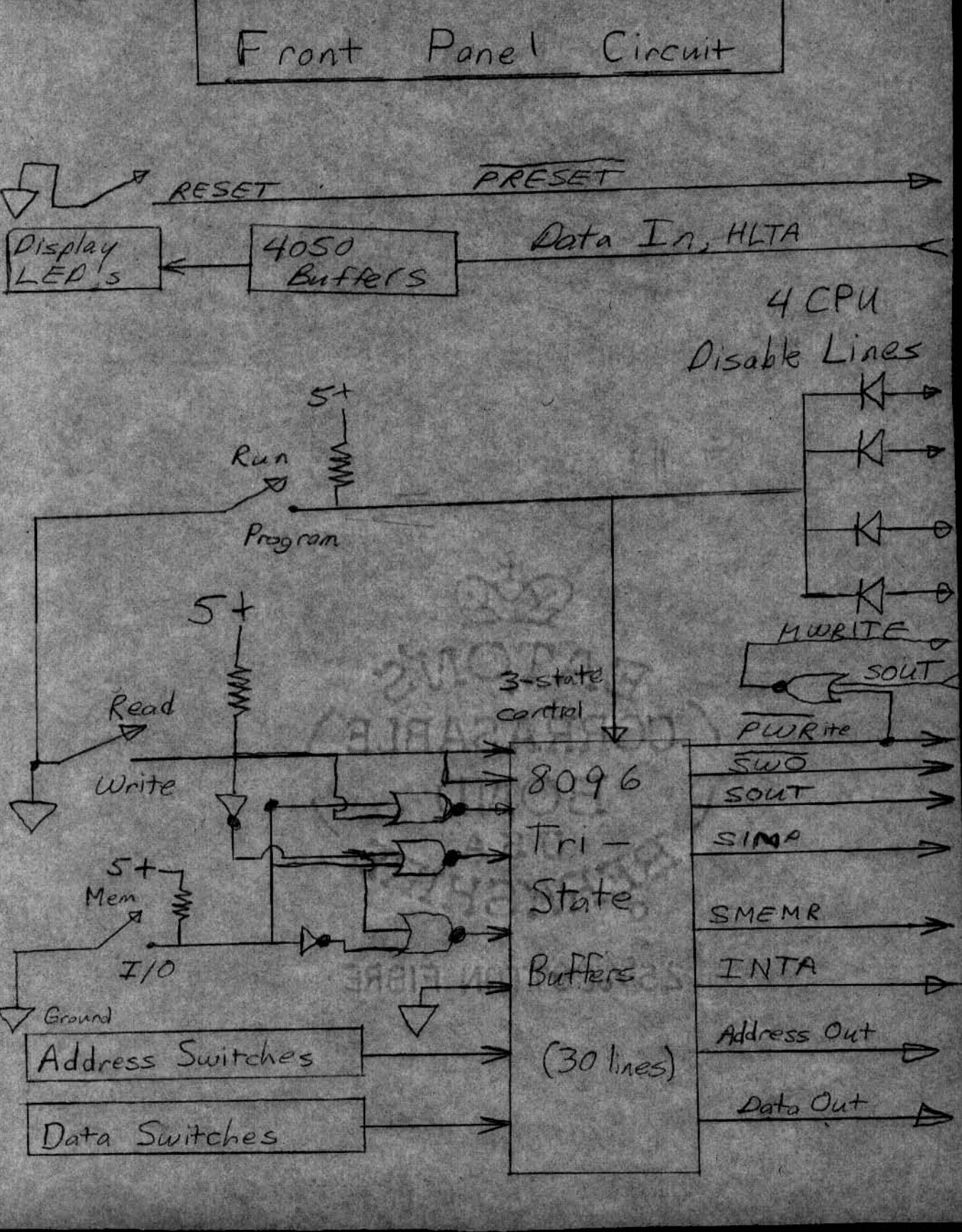
To the right is the circuit for this computer's front panel. This front panel can be used to deposit data in memory or read data from memory. Unlike most front panels, it is completely independant of the CPU; it does not need a CPU to work.

-How to operate the front panel.

The 8 black toggle switches are the DATA switches. The six black rotary switches are the ADDRESS switches.

This front panel has two basic modes of operation: RUN and PROGRAM. To program, set the RUN/PROGRAM switch on PROGRAM. Set the address switches to the desired address. The DATA LED's now show the contents of the memory location specified by the ADDRESS switches. To write into this memory location, put the desired write data on the DATA switches and press the WRITE switch. To run the program, set the RUN/PROGRAM switch on RUN.

Front Panel Circuit



8080 SOFTWARE

The 8080 microprocessor must be programmed with 8 bit numbers instructions (machine language). Humans cannot easily understand machine language. Assembly language, a language in which the 8 bit numbers are replaced 1 for 1 with easily understood command abbreviations, is easily understood by humans.

Because of this, I wrote a program on the school's computer that changes 8080 assembly code into 8080 machine code. Using this program, I wrote the following programs (for the 8080 processor).

Note: "OP CODE" stands for "operation code." Numbers under it are written in BASE 8, not BASE 2 or BASE 16.

```
ØLD,8080SIM
READY.
RN
? ØLD ADD
? LIST
LIST ØF     ADD

ADDRESS    ØP CØDE    ASSEMBLED

000000      041        LXI H
000001      100
000002      000
000003      176        MØV A M
000004      043        INX H
000005      206        ADD M
000006      043        INX H
000007      167        MØV M A
000010      166        HLT

LIST CØMPLETE
```

Start of Run 8080SIM

This program is an 8 bit binary adding program

Add

```
? ØLD SPREAD
? LIST
LIST ØF     SPREAD

ADDRESS    ØP CØDE    ASSEMBLED

000001      041        LXI H
000002      000
000003      000
000004      071        DAD SP
000005      042        SHLD
000006      100
000007      000
000010      166        HLT

LIST CØMPLETE
```

SP-READ

This program reads the Stack Pointer, and outputs its contents.

This program is a multi-digit binary adding program.

```
? ØLD ADDII
? LIST
LIST ØF     ADDII

ADDRESS   ØP CØDE    ASSEMBLED

000000      001       LXI B
000001      100
000002      000
000003      021       LXI D
000004      300
000005      000
000006      041       LXI H
000007      200
000010      000
000011      072       LDA
000012      000
000013      001
000014      057       CMA
000015      306       ADI
000016      201
000017      067       STC
000020      077       CMC
000021      062       STA
000022      001
000023      001
000024      012       LDAX B
000025      216       ADC M
000026      022       STAX D
000027      003       INX B
000030      023       INX D
000031      043       INX H
000032      072       LDA
000033      001
000034      001
000035      074       INP A
000036      362       JP
000037      021
000040      000
000041      166       HLT

LIST CØMPLETE
```

Addii

This promgram is an 8 bit binary multiplier

```
  ØLD MULT
? LIST
LIST ØF    MULT

ADDRESS    ØP CØDE    ASSEMBLED

000000     001        LXI B
000001     100
000002     000
000003     012        LDAX B
000004     003        INX B
000005     137        MØV B A
000006     012        LDAX B
000007     026        MVI D
000010     000
000011     041        LXI H
000012     000
000013     000
000014     037        RAR
000015     322        JNC
000016     021
000017     000
000020     031        DAD D
000021     353        XCHG
000022     051        DAD H
000023     353        XCHG
000024     267        ØRA A
000025     302        JNZ
000026     014
000027     000
000030     042        SHLD
000031     102
000032     000
000033     166        HLT

LIST CØMPLETE
```

Multiply

These programs loop for a certain amount of time, then halt.

```
  ØLD TIME
? LIST
LIST ØF     TIME

ADDRESS   ØP CØDE    ASSEMBLED

000000      041       LXI H
000001      000
000002      000
000003      021       LXI D
000004      001
000005      000
000006      031       DAD D
000007      322       JNC
000010      006
000011      000
000012      166       HLT

LIST CØMPLETE
```

Timer

```
? ØLD TIME2
? LIST
LIST ØF     TIME2

ADDRESS   ØP CØDE    ASSEMBLED

000000      076       MVI A
000001      000
000002      041       LXI H
000003      000
000004      000
000005      021       LXI D
000006      001
000007      000
000010      031       DAD D
000011      322       JNC
000012      010
000013      000
000014      306       ADI
000015      001
000016      322       JNC
000017      002
000020      000
000021      166       HLT

LIST CØMPLETE
```

Timer

```
IMPRØPER LØG IN, TRY AGAIN.
USER NUMBER: H5P3001/NS2X
TERMINAL:      30, TTYD.
RECØVER /SYSTEM: BAS,ØLD,8080SIM
READY.
NEW,MULT
READY.
AUTØ,0,1
00000 LXI B
00001 100
00002 000
00003 LDAX
00004  *DEL*
00003 LDAX B
00004 INX B
00005 MØV B A
00006 LDAX B
00007 MVI D
00008
00009
00010
      000
00011 LXI H
00012 000
00013 000
00014 RFC
00015 JNC
00016 021
00017 000
00018
00019
00020 DAD D
00021 XCHG
00022 DADH ←← H
00023 XCHG
00024 ØPA A
00025 JNZ
00026 014
00027 000
00028
00029
00030 SHLD
00031 102
00032 000
00033 HLT
00034  *DEL*
REP
READY.
```

How Assembly Language programs are entered.

L O G I C F A M I L Y C O M P A R I S O N

Charactaristic	7400	74LS00	74C00 (5 volt)
Propagation delay	11 nsec	10 nsec	40 nsec
Supply current/gate	2.0 mA	0.4 mA	0.0 mA
Power/gate	10. mW	2.0 mW	0.0 mW
Input current	1.6 mA	0.36 mA	0.00 mA
Drive current	16. mA	8. mA	1.75 mA
Fan out	10	20	50 or more

Fan out, 7400 to 74LS00 : 40

I used 7400 TTL for bus drivers on my CPU and memory boards because of this family's high drive current. I used 74LS00 TTL for all other requirements because of this family's low power consumption, high fan out, and low input current.

MOS logic : the 8080A and the 2102

The 8080A CPU chip and the 2102 memory chip both use MOS (metal oxide semiconductor) logic. MOS logic is similar in most respects to CMOS logic (74C00 logic). MOS logic consumes far more power than CMOS logic; it is also far more compact and far less expensive than CMOS logic. MOS logic is also less expensive and more compact than TTL logic (7400 and 74LS00 logic). For these reasons, MOS logic, rather than TTL or CMOS logic, is used in microcomputer CPU and memory chips.

GATES

A gate is any part or group of parts that can perform a boolean function. Here is a list of simple gate types and functions.

GATE NAME	OUTPUT
OR	$A+B$
AND	AB
NOR	$\overline{A+B}$
NAND	$\overline{AB}$
EXCLUSIVE OR	$\bar{A}B+A\bar{B}$
EXCLUSIVE NOR	$\bar{A}\bar{B}+AB$
NOT (INVERTER)	$\bar{A}$

The first four types of gates can have two or more inputs. A gate that performs the function $Q=A+B+C+D+E$ is considered an OR gate.

These gates can be used together to perform complex boolean functions.

CONSTRUCTION OF GATES

Gates can be constructed from switches, relays, diodes, transistors and many other devices. Compact, ready-made gates (IC's) are available at very low prices. These IC's (integrated circuits) come in three main types, TTL, CMOS, and MOS.

Diodes can be used to construct AND and OR gates. Buffer amplifiers must be used with these gates. (Figure A)

Transistors and resistors can be used to construct all the types of gates. The basic gates that can be constructed with bipolar transistors are the NOT, NAND, and NOR gates. By using NOT gates, NAND and NOR can be changed to AND and OR gates. (Figure B)

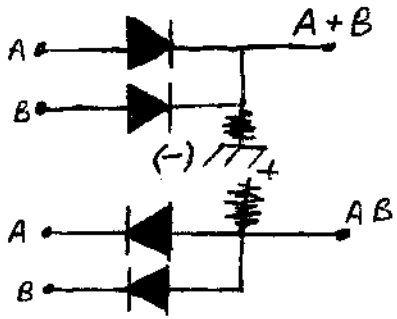
A TTL (transistor-transistor logic) IC can contain any type of gate. In this type of gate, many bipolar transistors are built into one compact package. These gates are very popular, but they are being replaced in many applications by MOS and CMOS logic.

MOS (metal-oxide semiconductor) are gaining significance. P-MOS is the most popular type of MOS. One MOS device contains many MOS transistors built into one compact package. These IC's can contain any type of gate, but only complex functions have become popular.

CMOS (complementary metal-oxide semiconductor) IC's contain both P-channel and N-channel MOS transistors. They consume very little power. Figure C is an inverter (NOT gate) that is contained in all CMOS gates. Other types of gates are built inside the IC out of inverters.

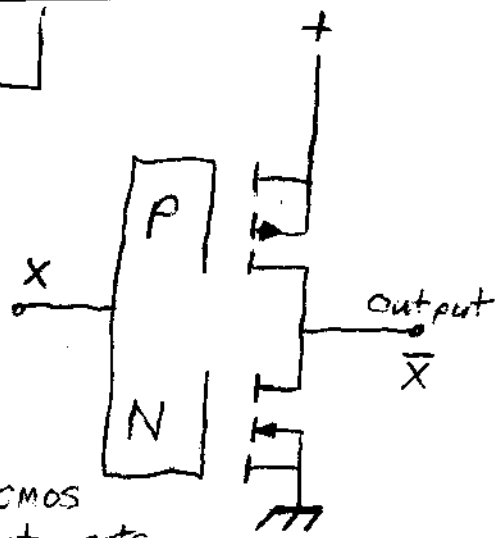
TTL, MOS, and CMOS logic are the main types of logic in use today. Prices have been dropping steadily, while quality has climbed.

A



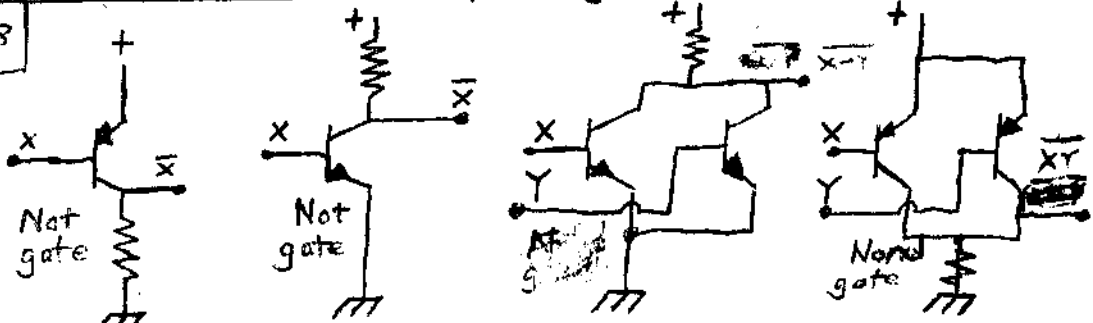
Diode gates

C



CMOS not gate

B



Nor gate

$\overline{\overline{X+Y}} = XY$ Nor1 to And

$\overline{(\bar{X})+(\bar{Y})} = X+Y$ Nan1 to Or

$\overline{\bar{X}\bar{Y}} = X+Y$ Nor to Or

$\overline{(\bar{X})(\bar{Y})} = XY$ Nor to And