INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) - P.O.B. 586 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONES: 224281/2/3/4/5/6
CABLE: CENTRATOM - TELEX 460392-1

SMR/90 - 14

COLLEGE ON MICROPROCESSORS:

TECHNOLOGY AND APPLICATIONS IN PHYSICS

7 September - 2 October 1981

LOGIC AND ARITHMETIC

A. WERBROUCK

Istituto di Scienza
dell'Informazione
Corso Massimo d'Azeglio, 42
Torino, Italy

BOOLEAN ALGEBRA AND LOGIC CIRCUITS

NECESSARY TO UNDERSTAND:
- INTERNAL FUNCTIONING
- INTERFACING WITH PERIPHERALS
- APPLICATIONS (REPLACE HARD-WIRED LOGIC)

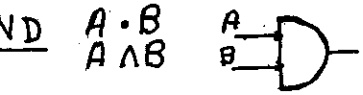
AS FORMAL MATHEMATICAL SYSTEM, BOOLEAN ALGEBRA HAS:

1. SET OF ELEMENTS (CONSTANTS) (0, 1)
   LOGIC-1 IS AFFIRMATION (YES, TRUE
   LOGIC-0 IS NEGATION (NO, FALSE,

LOGIC VARIABLES (LETTERS) MUST HAVE 0 OR 1 VALUE

2. OPERATIONS-SYMBOLS 3. POSTULATES TRUTH TABLE

AND $A \cdot B$ $A \wedge B$



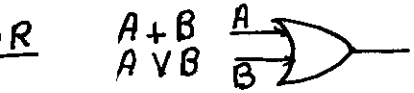
A \ B	0	1
0	0	0
1	0	1

PRODUCT

LOGICAL MULTIPLICATION

GENERALIZED $A \cdot B \cdot C \ldots$

OR $A+B$ $A \vee B$



A \ B	0	1
0	0	1
1	1	1

SUM

LOGICAL ADDITION

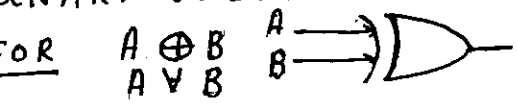
GENERALIZED $A+B+C \ldots$.

NOT= COMPLEMENTATION, NEGATION, INVERSION → $\overline{A}$ OR A'

A	$\overline{A}$
0	1
1	0

UNARY OPERATION

EOR $A \oplus B$ $A \vee B$



A \ B	0	1
0	0	1
1	1	0

COMPOSITE $A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$

GENERALIZED = $A \oplus B \oplus C \ldots$ = EVEN PARITY

DIGITAL ELECTRONICS NECESSARY FOR UNDERSTANDING FUNCTIONING OF DIGITAL SYSTEMS AND SUB-SYSTEMS.

1. BASED ON DISCRETE (SEPARATED, DISTINCT) VALUES OF VOLTAGE, CURRENT, FREQUENCY, OR PHASE OF NATURALLY CONTINUOUS PHYSICAL VARIABLES COMPOSED OF ONE OR MORE SIGNALS.

   SIGNAL = AMPLITUDE * COSINE (2π * FREQUENCY * TIME + PHASE)

2. NORMALLY TWO LEVELS ARE DEFINED OF BINARY SYSTEM YES-NO, ON-OFF, HIGH-LOW, POS-NEG, ONE-ZERO, RIGHT-LEFT TRUE-FALSE, ETC.

3. INTERESTING STATES CORRESPOND TO BINARY NUMBER (0-1). REPRESENTATION WITH EXTENSIONS (X = DON'T CARE, TRI-STATE, ETC.) AS ENABLE-DISABLE.

PRIMITIVE FUNCTIONS FOR BUILDING ALL OTHERS:

OF ONE VARIABLE A: $F(A) = A$ (TRUE), $\overline{A}$ (COMPLEMENT), 0 (ZERO), 1 (ONE)

AS BOOLEAN FUNCTION → A	$F(A)=A$	$F(A)=\overline{A}$	$F(A)=0$	$F(A)=1$
AS TRUTH TABLE → 0	0	1	0	1
1	1	0	0	1

NUMBER OF FUNCTIONS

OF N VARIABLES:

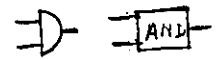
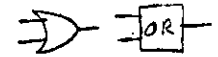
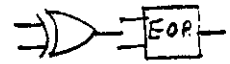
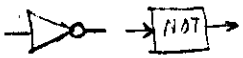
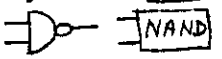
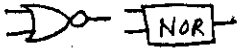
(NUMBER OF VALUES OF F) = (NUMBER OF VALUES OF VARIABLE) ^ (NUMBER OF VARIABLES)

w^n = NUMBER OF CELLS IN TRUTH TABLE

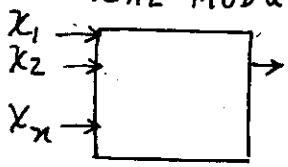
FOR EACH COVERING OF THE CELLS, A FUNCTION WHICH GIVES A DISTINCT SET OF VALUES. WRITABLE AS A BINARY NUMBER OF w^n BITS. FOR THE INTERESTING CASE ($n=2$): 16 FUNCTIONS

A=0 B=0	A=0 B=1	A=1 B=0	A=1 B=1	COVERING ↓ FUNCTION
0	0	0	0	ZERO
0	0	0	1	$A \cdot B$
0	0	1	0	$A \cdot \overline{B}$
0	0	1	1	A
0	1	0	0	$\overline{A} \cdot B$
0	1	0	1	B
0	1	1	0	$\overline{A} \cdot B + A \cdot \overline{B} = A \oplus B$
0	1	1	1	$A + B$
1	0	0	0	$\overline{A} \cdot \overline{B} = \overline{A+B}$
1	0	0	1	$A \cdot B + \overline{A} \cdot \overline{B} = \overline{A \oplus B}$
1	0	1	0	$\overline{B}$
1	0	1	1	$A + \overline{B}$
1	1	0	0	$\overline{A}$
1	1	0	1	$\overline{A} + B$
1	1	1	0	$\overline{A \cdot B} = \overline{A} + \overline{B}$
1	1	1	1	1

IN PRACTICE, WE CHOOSE THE FOLLOWING SUBSET.

NAME -	EXPRESSION -	SYMBOLS
AND:	$A \cdot B$	
OR:	$A+B$	
EOR	$A \oplus B$	
NOT	$\overline{A}$	
NAND	$\overline{A \cdot B}$	
NOR	$\overline{A+B}$	

LOGICAL MODULE OR SUB-UNIT



COMBINATIONAL NETWORK
NO INTERNAL MEMORY

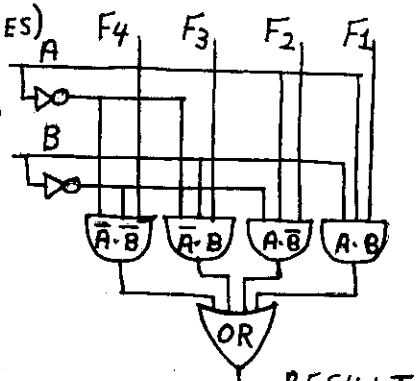
CAN BE DESCRIBED BY TRUTH TABLE, LOGIC EQUATIONS, EVEN MEMORY CONTENTS

EXAMINE COMPLETE COMBINATORIAL LOGIC CIRCUIT OF TWO VARIABLES (16 FUNCTIONS)
RESULT = $F_4 \cdot \bar{A}\,\bar{B} + F_3 \cdot \bar{A}\,B + F_2 \cdot A\,\bar{B} + F_1 \cdot A \cdot B$

TRUTH TABLE (42 LINES)
X = DON'T CARE

F_4	F_3	F_2	F_1	A	B	RESULT
0	0	0	0	X	X	0 ZERO
0	0	0	1	0	X	0 $A \cdot B$
				X	0	0
				1	1	1
0	0	1	0	0	X	0 $A \cdot \bar{B}$
				X	1	0
				1	0	1
0	0	1	1	0	X	0 A
				1	X	1
0	1	0	0	1	X	0 $\bar{A} \cdot B$
				X	0	0
				0	1	1
0	1	0	1	X	0	0 B
				X	1	1
0	1	1	0	0	0	$\bar{A} \cdot B + A \cdot \bar{B}$
				0	1	1 = EOR
				1	0	1
				1	1	0
0	1	1	1	0	0	0 $A + B$
				1	X	1
				X	1	1
1	0	0	0	1	X	0 $\bar{A} \cdot \bar{B}$
				X	1	0
				0	0	1
1	0	0	1	0	0	1 $\bar{A} \cdot \bar{B} + A \cdot B =$
				0	1	0 EQUIVALENCE
				1	0	0
				1	1	1

DO THE REST



LOOK-UP TABLE
IN NORMAL MEMORY

ADDRESS = ARGUMENT	CONTENT = RESULT	
64 WORDS		
0 0 0 0 0 0	0	0 ZERO
	0 1	0
	1 0	0
	1 1	0
0 0 0 1 0 0	0	0 $A \cdot B$
	0 1	0
	1 0	0
	1 1	1
0 0 1 0 0 0	0	0 $A \cdot \bar{B}$
	0 1	0
	1 0	1
	1 1	0

DO THE REST AS AN EXERCISE

BOOLEAN ALGEBRA THEOREMS

1a. $\bar{0} = 1$	1b. $\bar{1} = 0$	"NOT" POSTULATE
2a. $A+0=A$	2b. $A \cdot 1 = A$	AND, OR
3a. $A+1=1$	3b. $A \cdot 0 = 0$	AND, OR
4a. $A+A=A$	4b $A \cdot A = A$	IDEMPOTENT LAW
5a. $A+\bar{A}=1$	5b $A \cdot \bar{A} = 0$	REFLECT COMPLEMENTARY PROPERTY OF VARIABLES
6. $\overline{(\bar{A})} = A$		INVOLUTION (FROM 1)
7a. $A+B=B+A$	7b $A \cdot B = B \cdot A$	COMMUTATIVE
8a. $A+AB=A$	8b $A(A+B)=A$	ABSORPTION FOR SIMPLIFICATION
9a. $A+\bar{A}B=A+B$	9b $A(\bar{A}+B)=AB$	SIMPLIFICATION
10a. $\overline{(A+B)} = \bar{A} \cdot \bar{B}$	10b $\overline{(AB)} = \bar{A}+\bar{B}$	DEMORGAN'S LAW
11a. $(A+B)+C = A+(B+C) = A+B+C$	11b $(AB)C = A(BC) = ABC$	ASSOCIATIVE

12a $A + BC = (A+B)(A+C)$ 12b $A(B+C) = AB + AC$ DISTRIBUTIVE

PRINCIPLE OF DUALITY ($a \leftrightarrow b$, AND $\leftrightarrow$ OR, $1 \leftrightarrow 0$)

ALL CAN BE PROVEN BY PERFECT INDUCTION, THAT IS BY SUBSTITUTING ALL POSSIBLE VALUES.

GENERALIZED FORMS:
DEMORGANS

$$\frac{\overline{A+B+C\ldots}}{\overline{ABC\ldots}} = \frac{\bar{A} \cdot \bar{B} \cdot \bar{C}}{\bar{A}+\bar{B}+\bar{C}}$$

DISTRIBUTIVE

$$A + BCD\ldots = (A+B)(A+B)(A+D)\ldots$$

$$A(B+C+D\ldots) = AB + AC + AD\ldots$$

TRANSFORMATIONS

LOGIC EQUATION → TRUTH TABLE
SUBSTITUTE VALUES OF INPUT VARIABLES INDICATED BY INDICES OF TRUTH TABLE

Row	A	B	C	F
	0	0	1	

$F(A,B,C) = (A + \overline{B}) \cdot C \rightarrow (0+1) \cdot 1 = 1$

MORE COMMON TRANSITION: TRUTH TABLE → LOGIC EQUATIONS

WRITE TRUTH TABLE AS KARNAUGH MAP

2-VARIABLE

A \ B	0	1
0	F(0,0)	F(0,1)
1	F(1,0)	F(1,1)

3 VARIABLE

A \ BC	00	01	11	10
0	F(0,0,0)	F(0,0,1)	F(0,1,1)	F(0,1,0)
1	F(1,0,0)	F(1,0,1)	F(1,1,1)	F(1,1,0)
	$\overline{C}$	C	C	$\overline{C}$

(NOTE WRAPAROUND PROPERTY)

AND THEN COVER ONES WITH SUM OF PRODUCTS

DOUBLE COVERING IS PERMITTED (A $\overline{B}$ C)

A \ BC	00	01	11	10
0	1	1	0	1
1	0	1	1	X

DON'T CARE COVERING OPTIONAL $\overline{B}C$ AC

$F = AC + \overline{B}C + \overline{A}\,\overline{C}$ (WRAPAROUND TERM)

4-VARIABLE KARNAUGH MAP IS REALLY A TORUS DUE TO DOUBLE WRAPAROUND.

AB \ CD	00	01	11	10
00	0	X	0	1
01	X	1	0	1
11	0	0	X	X
10	0	1	0	1

$\overline{B}\,\overline{C}\,D$ $C\,\overline{D}$

$F = \overline{A}B\overline{C} + \overline{B}\,\overline{C}D + C\overline{D}$.

MINIMAL SUM

AB \ CD	00	01	11	10
00	0	X	0	1
01	X	1	0	1
11	0	0	X	X
10	0	1	0	1

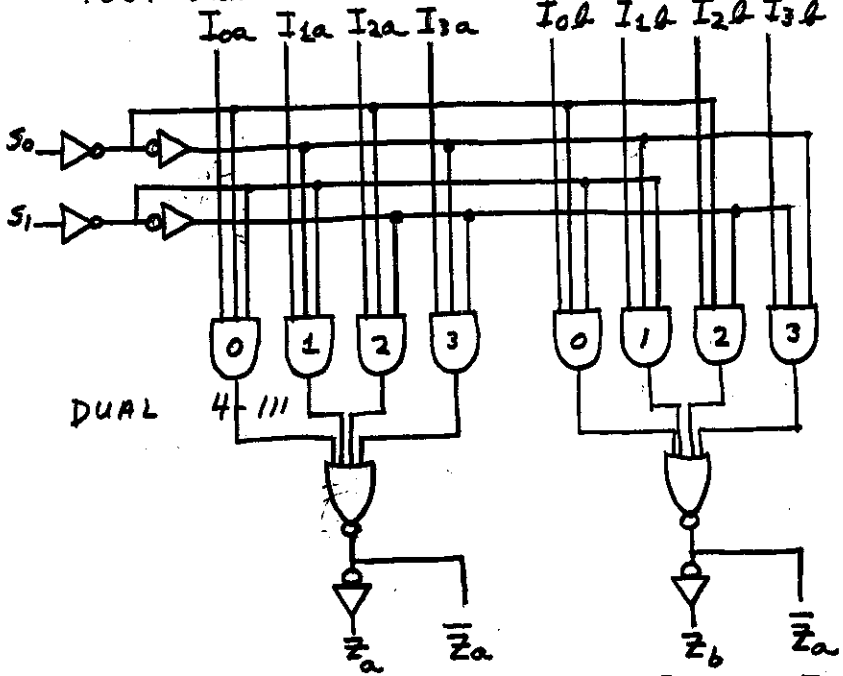
COVER 0'S $\overline{C}\,\overline{D}$ $C\overline{D}$

$\overline{F} = \overline{C}\,\overline{D} + CD + AB\overline{C}$

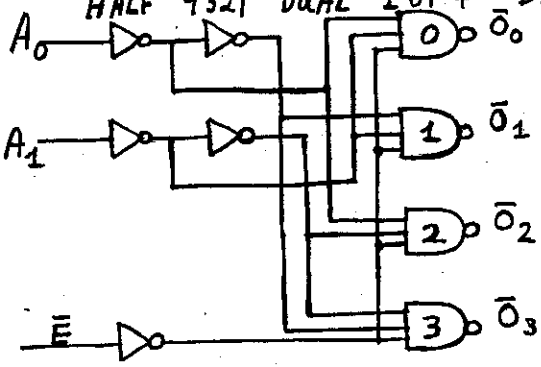
$F = \overline{(\overline{C}\,\overline{D})} \cdot \overline{C\overline{D}} \cdot \overline{(AB\overline{C})}$

MINIMAL PRODUCT → $F = (C+D)(\overline{C}+\overline{D})(\overline{A}+\overline{B}+C)$

9309 DUAL 4-INPUT MULTIPLEXER



HALF 9321 DUAL 1-OF 4 DECODER-DEMULTIPLEXER



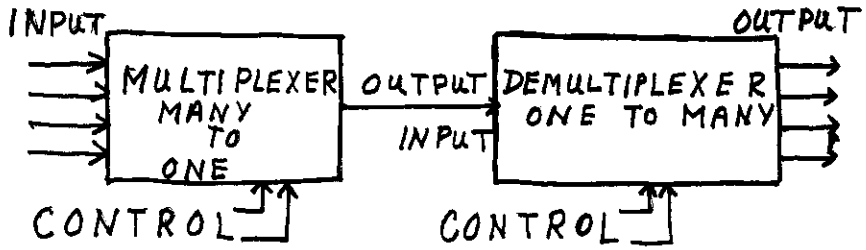
DECODER TRUTH TABLE OF MULTIPLEXER

S_1	S_0	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

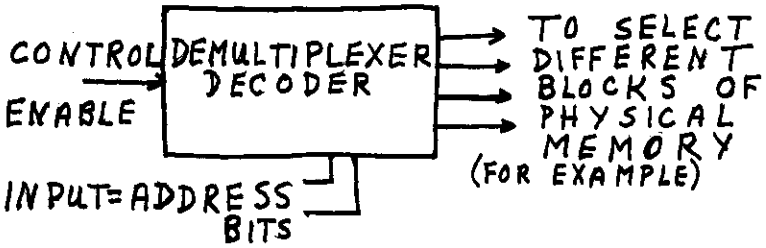
TRUTH TABLE OF DECODER

A_1	A_0	O_0	O_1	O_2	O_3
0	0	$\overline{E}$	H	H	H
0	1	H	$\overline{E}$	H	H
1	0	H	H	$\overline{E}$	H
1	1	H	H	H	$\overline{E}$

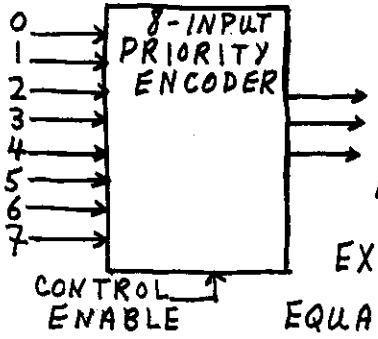
USE OF MULTIPLEXER-DEMULTIPLEXER



USE OF DEMULTIPLEXER AS DECODER

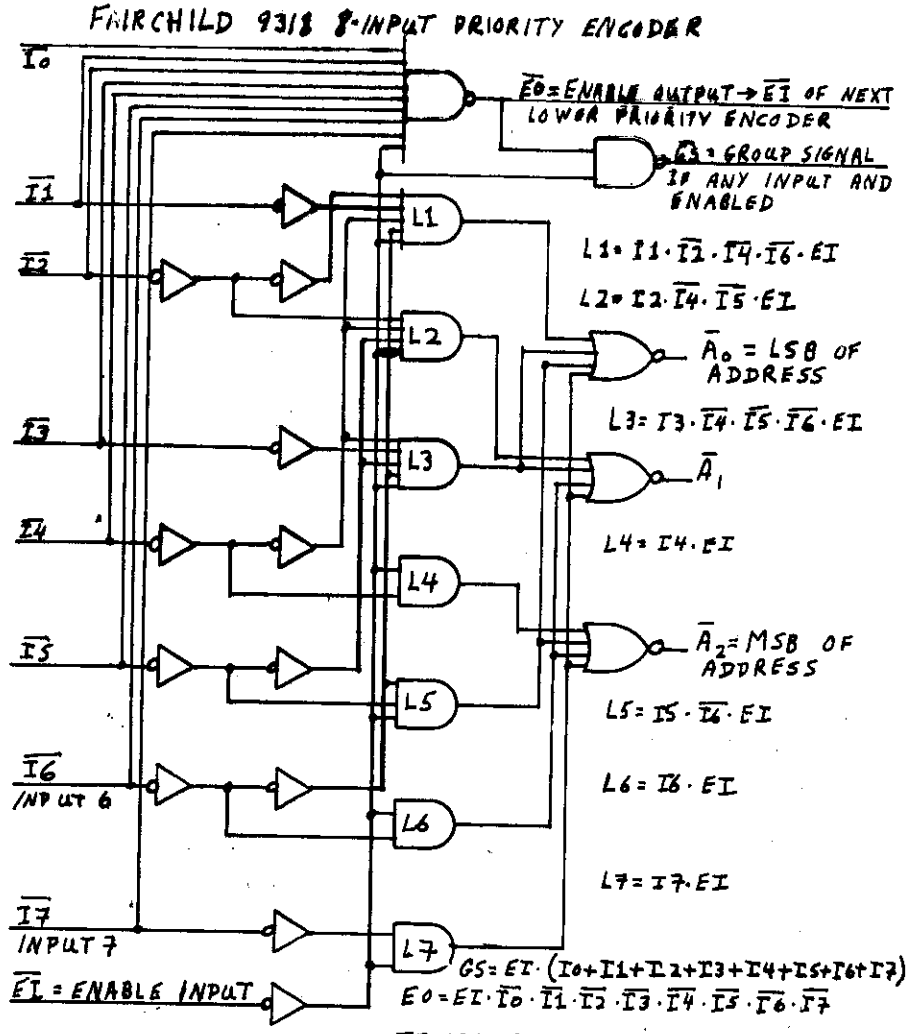


TO ENCODE - USE PRIORITY ENCODER TO AVOID AMBIGUITY IF MORE THAN ONE INPUT LINE IS ASSERTED (= TRUE, 1)



EXERCISE: VERIFY BOOLEAN EQUATIONS OF FAIRCHILD 9318 CASCADABLE 8-INPUT PRIORITY ENCODER ON NEXT PAGE.

FAIRCHILD 9318 8-INPUT PRIORITY ENCODER



TRUTH TABLE

$\overline{EI}$	$\overline{I0}$	$\overline{I1}$	$\overline{I2}$	$\overline{I3}$	$\overline{I4}$	$\overline{I5}$	$\overline{I6}$	$\overline{I7}$	$\overline{GS}$	$\overline{A}_0$	$\overline{A}_1$	$\overline{A}_2$	$\overline{EO}$	
H	X	X	X	X	X	X	X	X	H	H	H	H	H	(DISABLED)
L	H	H	H	H	H	H	H	H	H	H	H	H	L	(NO ACTIVE INPUT)
L	X	X	X	X	X	X	X	L	L	L	L	L	H	
L	X	X	X	X	X	X	L	H	L	H	L	L	H	
L	X	X	X	X	X	L	H	H	L	L	H	L	H	
L	X	X	X	X	L	H	H	H	L	H	H	L	H	
L	X	X	X	L	H	H	H	H	L	L	L	H	H	
L	X	X	L	H	H	H	H	H	L	H	L	H	H	
L	X	L	H	H	H	H	H	H	L	L	H	H	H	
L	L	H	H	H	H	H	H	H	L	H	H	H	H	

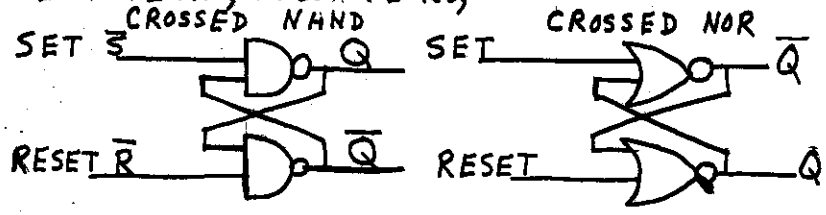
SEQUENTIAL LOGIC CIRCUITS

CONTAIN MEMORY OF PREVIOUS INPUTS

CALLED "STATE" IN COMPUTER JARGON

POSITIVE LOGIC: 1 = HIGH VOLTAGE = H
0 = LOW VOLTAGE = L

USE BISTABLE ELEMENT (FLIP-FLOP) TO COMPOSE REGISTERS, COUNTERS,



TRUTH TABLE-CONTROL L

$\bar{R}$	$\bar{S}$	Q	$\bar{Q}$
H	H	NO CHANGE	
L	H	L	H
H	L	H	L
L	L	H	H

INDETERMINATE

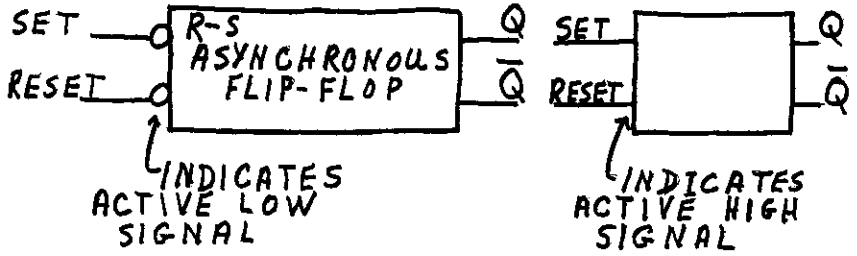
TRUTH TABLE-CONTROL H

R	S	Q	$\bar{Q}$
H	H	L	L
L	H	H	L
H	L	L	H
L	L	NO CHANGE	

INDETERMINATE

NOTE FEED BACK FROM OUTPUT TO GATE INPUT TO MAINTAIN STATE (IF VOLATILE)

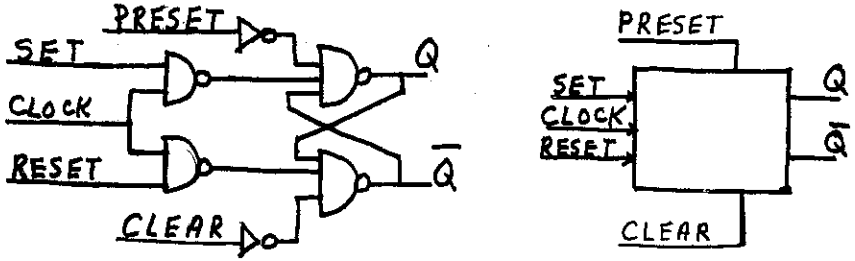
BLOCK DIAGRAMS:



WE CONTINUE ONLY WITH CROSSED NAND TYPE

SYNCHRONOUS FLIP-FLOP

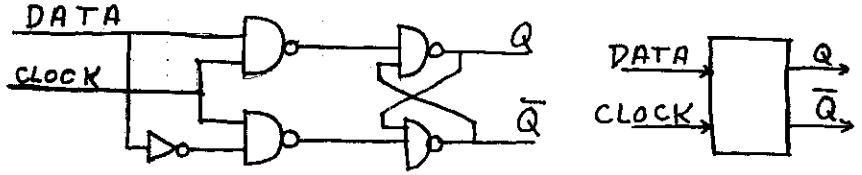
ADD SYNCHRONOUS OR CLOCKED INPUT



TRUTH TABLE FOR PRESET = CLEAR = LOW

R	S	Q	$\bar{Q}$ (AFTER CLOCK DESCENT)
H	H	INDETERMINATE - TO BE REMOVED BY JK	
H	L	L	H
L	H	H	L
L	L	Q_0	$\bar{Q}_0$ = Q, $\bar{Q}$ VALUES BEFORE CLOCK

CLOCKED D FLIP-FLOP OR DATA LATCH HAS NO INDETERMINATE STATE



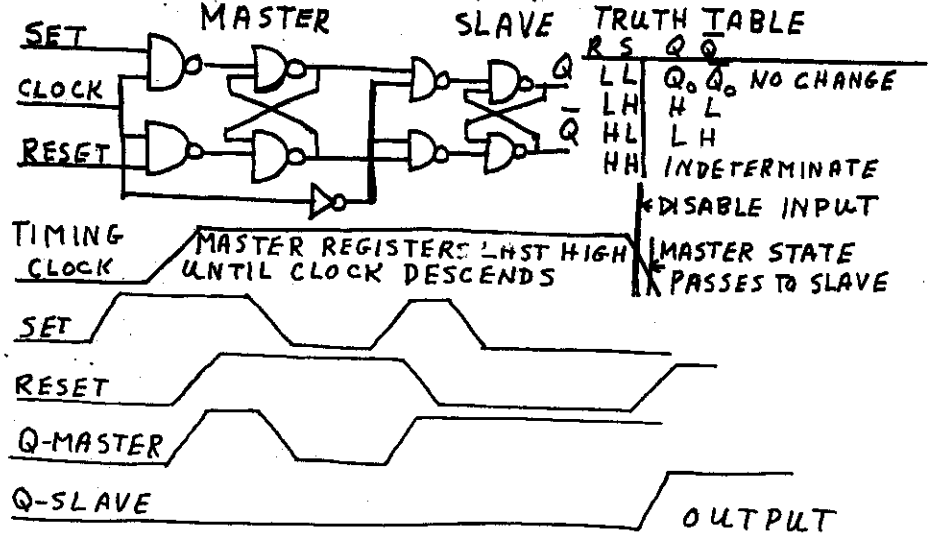
TRUTH TABLE

DATA WHEN CLOCK DESCENDS	Q	$\bar{Q}$ AFTER CLOCK DESCENDS
H	H	L
L	L	H

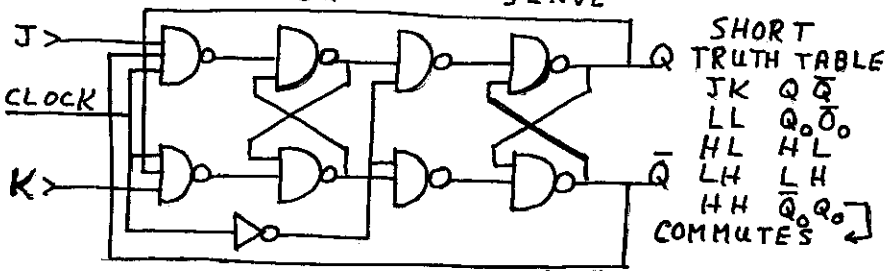
NOTE: TRANSPARENT WHEN CLOCK HIGH

CANNOT BE USED FOR SHIFT REGISTER OR REGISTER WHICH MUST MEMORIZE RESULT DEPENDING ON PREVIOUS RESULT A+B → A.

1. REMOVE TRANSPARENCY WITH MASTER-SLAVE STRUCTURE (MS)



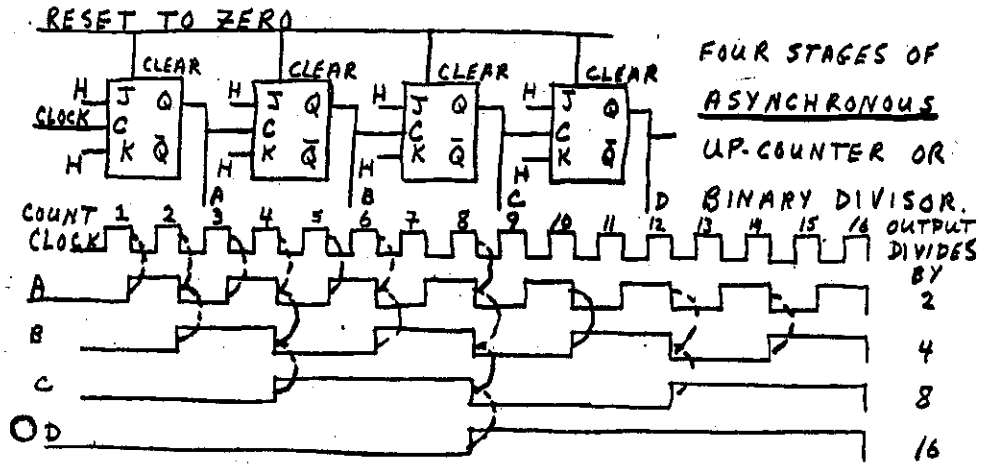
2. REMOVE INDETERMINATE STATE WITH JK MS SECTION-MASTER SLAVE



LONG TRUTH TABLE - Q_0 $\bar{Q}_0$ = OUTPUT BEFORE CLOCK RISE

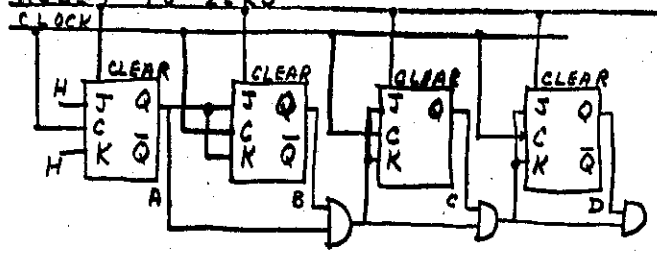
J	$\bar{Q}_0$	$J\cdot\bar{Q}_0$	K	Q_0	$K\cdot Q_0$	Q	$\bar{Q}$	RESULT
L	L	L	L	H	L	H	L	NO CHANGE
L	L	L	H	H	H	L	H	CHANGE
H	L	L	L	H	L	H	L	NO CHANGE
H	L	L	H	H	H	L	H	CHANGE - COMMUTES
L	H	L	L	L	L	L	H	NO CHANGE
L	H	L	H	L	L	L	H	NO CHANGE
H	H	H	L	L	L	H	L	CHANGE
H	H	H	H	L	L	H	L	CHANGE - COMMUTES

COUNTERS - ALL BASED ON J-K MASTER-SLAVE FLIP-FLOP



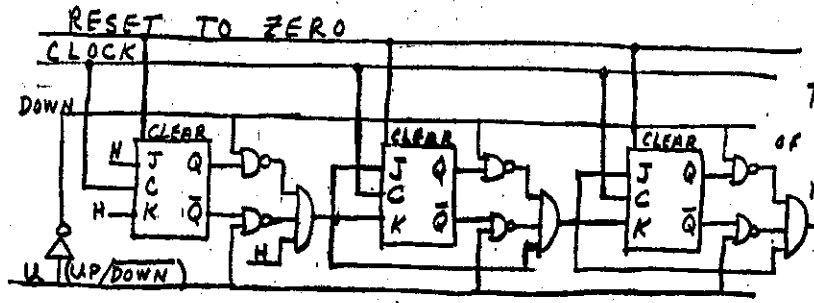
FOUR STAGES OF ASYNCHRONOUS UP-COUNTER OR BINARY DIVISOR.

SYNCHRONOUS COUNTER APPLIES CLOCK TO EACH FLIP-FLOP



FOUR STAGES OF SYNCHRONOUS UP COUNTER OR BINARY DIVISOR

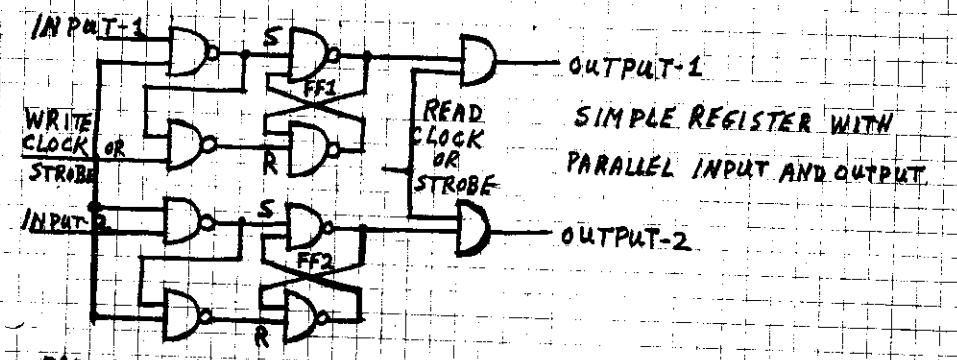
REVERSIBLE COUNTER IS SYNCHRONOUS COUNTER WITH COMPLETE SYMMETRIC PRE-CONDITIONING OF J-K INPUTS OF NEXT HIGHER ORDER STAGES.

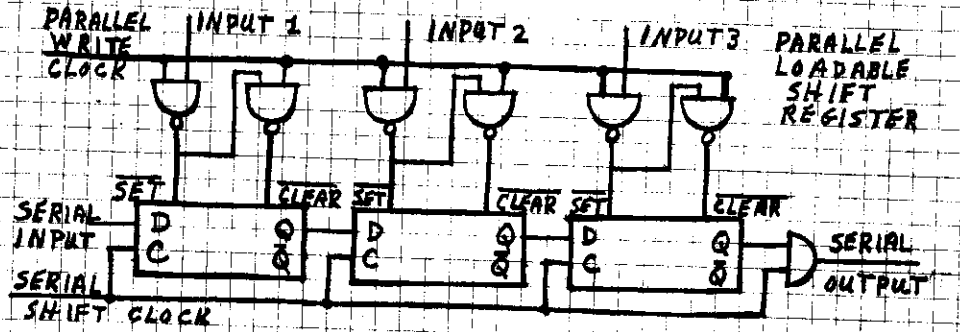


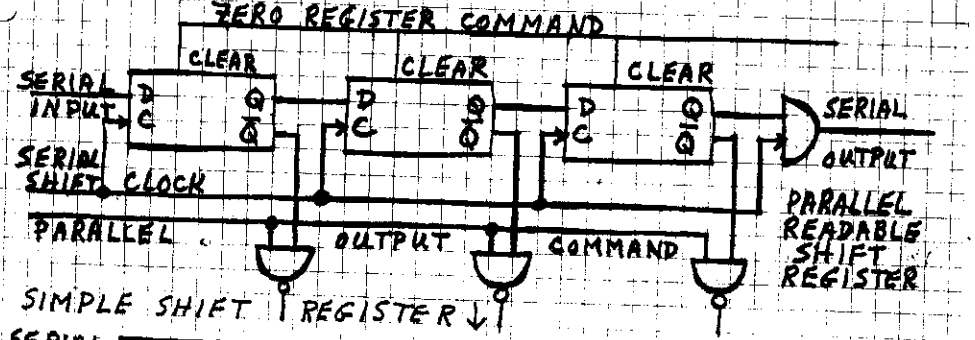
THREE STAGES OF SYNCHRONOUS REVERSIBLE COUNTER.

COMMUTE FF_{N+1} IF AT ALL $N, N-1, \ldots$, $\bar{Q}u \cdot \overline{Q\bar{u}} = \overline{Q \oplus u}$.

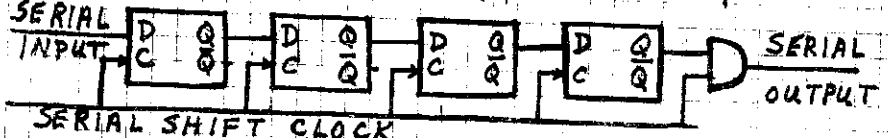
REGISTERS; USED WIDELY FOR FAST, LOW CAPACITY MEMORY, CONTAIN
1) MECHANISM FOR WRITING
2) MECHANISM FOR READING
3) MEMORY ELEMENT





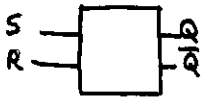


SIMPLE SHIFT REGISTER



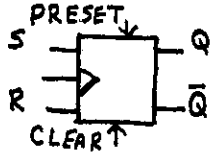
SUMMARY OF FLIP-FLOPS AND USES

ASYNCHRONOUS R-S



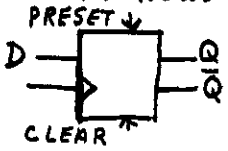
INTERNAL STATE VARIABLES

SYNCHRONOUS R-S



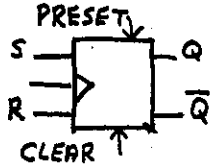
CLOCKED STATE VARIABLES

SYNCHRONOUS "D" SIMPLE LATCH

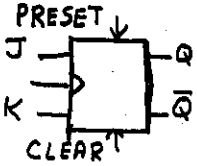


DATA REGISTRATION WHERE TRANSPARENCY IS NO PROBLEM

MASTER-SLAVE

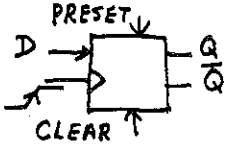


J-K MASTER SLAVE



COUNTERS

EDGE-TRIGGERED "D"



DATA REGISTERS
SHIFT REGISTERS
ARITHMETIC REGISTERS (A+B→A)

NUMBER REPRESENTATION

1 HUMAN DECIMAL: ± DDDD WHERE D IS DECIMAL DIGIT. EXAMPLES +5, −10 (SIGN AND MAGNITUDE)

2. COMPUTER BINARY USING ONLY TWO ELEMENTS (0,1) LET 0 = PLUS (+) AND 1 = MINUS (−)

LET MAGNITUDE BE IN POSITIONAL NOTATION USING BASE (RADIX) = 2 +5 = 00101, −10 = 11010

$5 = 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 0 + 4 + 0 + 1$

WHEN WE ADD AND SUBTRACT WE NORMALLY TREAT THE SIGN SEPARATELY.

HOWEVER, USING TEN'S COMPLEMENT NOTATION WE CAN ADD AS USUAL AND SUBTRACT BY ADDING THE TEN'S COMPLEMENT

```
 89 →  089   = 89          25 → 025
-25 →  175   =200-25       -89 → 111
       1064                    0136 =200-64=-64 (mod 200)
CARRY ↑ SIGN ↑ MAGNITUDE   CARRY ↑ SIGN ↑
```

```
 99 → 099  OVERFLOW        -99 → 101  OVERFLOW
+44   044                 +(-44) → 156
      0143                       1057 = −143
```

CORRECT SIGN IS IN CARRY POSITION WHEN OVERFLOW IS ASSERTED BY CS⊕CM=1 WHERE CS IS CARRY FROM SIGN POSITION, AND CM IS CARRY FROM MOST SIGNIFICANT DIGIT. BINARY ADDITION IS USED IN THE SIGN COLUMN AND DECIMAL ADDITION IN MAGNITUDE PART.

NOW SAME APPROACH APPLIED TO BINARY ARITHMETIC IS ESPECIALLY ATTRACTIVE

BECAUSE NOW BOTH SIGN AND MAGNITUDE PARTS OBEY BINARY ADDITION.

```
 89 = 0101 1001          -89 = 1010 0111
 25 = 0001 1001          -25 = 1110 0111
    0 0111 0010 = 114       1 1000 1110 = −114
```

```
 89  0101 1001          -89  1010 0111   OVERFLOW
-89  1010 0111          -89  1010 0111   RESULT
   0 0000 0000 = 0          1 0100 1110
                    NEGATE → 0101 0010 = 178
```

```
 89  0101 1001  OVERFLOW     64  0100 0000  OVERFLOW
 89  0101 1001               64  0100 0000
   0 1011 0010 = 178            0 1000 0000
```

```
-64  1100 0000   NO OVERFLOW FOR (-64)+(-64) ?
-64  1100 0000   OVERFLOW FOR (+64)+(+64) ?
   1 1000 0000
```

CONFIRMATION OF VALUE OF 1000 0000

```
LARGEST POSITIVE NUMBER = 0111 1111
        TAKE NEGATIVE     1000 0001
SUBTRACT ONE ≡ ADD  -1    1111 1111
                        1 1000 0000  NO OVERFLOW
```

1000 0000 IS VALID 2C NEGATIVE NUMBER WITH MAGNITUDE ONE LARGER THAN LARGEST POSITIVE, ALSO IT IS OWN 2C COMPLEMENT

SMALLER MAGNITUDE NEGATIVE NUMBERS ARE FORMED BY ADDING POSITIVE CONTRIBUTIONS WHICH APPEAR PLAINLY IN LOWER PORTION. THUS

$$\text{NUMBER(2C)} = \underbrace{-B_{w-1}*2^{w-1}}_{\text{SIGN}} + \sum_{i=0}^{w-2} B_i * 2^i$$

BY ADDING NUMBER AND ITS 2C COMPLEMENT WE FOUND THAT

$N + (-N) = (C=1) +$ ALL ZEROS $\equiv 2^w$ WHERE w = WIDTH OF REGISTER IN BITS

LET:

MAGNITUDE OF POSITIVE NUMBER = $0 \le a \le 2^{w-1}-1$

THEN NEGATIVE NUMBER = $2^w - a$

WITH MAGNITUDE: $0 < |2^w - a| \le 2^{w-1}$.

FOR 8-BIT MICROPROCESSOR

$0 \le a \le 127$; $\quad 0 < |256-a| \le 128$

REMEMBER THAT LATTER EXPRESSION HAS SIGN INTEGRATED (EMBEDDED) INTO REPRESENTATION SO THAT SIGNS AND MAGNITUDES CAN BE TREATED SIMULTANEOUSLY IN SAME MANNER.

THEN HAVE WE LOST DISTINCTION BETWEEN

SIGN AND MAGNITUDE?

ANSWER: THE COMPUTER HAS LOST IT

WE (COMPUTER PROGRAMMER) MUST

PROPERLY INTERPRET RESULTS!

RESULTS ARE CONTENTS OF REGISTERS.

TWO CASES:

1. NUMBERS ARE SIGNED (FIRST BIT IS SIGN)
2. NUMBERS ARE UNSIGNED (ONLY MAGNITUDE)

NOTE POSITIVE BYTE — 0000 0000 | 0101 1101 — IN 16 BITS

NEGATIVE BYTE — 1111 1111 | 1010 0011 — IN 16 BITS

SIGN MUST BE EXTENDED ↑

RESULT INTERPRETATION IN TWO'S COMPLEMENT ARITHMETIC

$$N = -B_{w-1} * 2^{w-1} + \sum_{i=0}^{w-2} B_i * 2^i$$

LET: $|A| = a$ $\quad -|A| = 2^w - a$

$|B| = b$ $\quad -|B| = 2^w - b$

w = WORD WIDTH

$m = w - 1$

ADDITION OF SIGNED INTEGERS A+B

SA,SB	CS,SR = 00	01	10	11
++ 00	OK $a+b<2^m$	OV $a+b \ge 2^m$		
+- 01		OK $a<b$	OK $a \ge b$	
-+ 10		OK $a>b$	OK $a \le b$	
-- 11			OV $a+b>2^m$	OK $a+b \le 2^m$

IF OV = CS ⊕ CM NOT AVAILABLE THEN USE

$OV = ADD \cdot (SA \cdot SB \cdot \overline{SR} + \overline{SA} \cdot \overline{SB} \cdot SR) + SUB \cdot (SA \cdot \overline{SB} \cdot \overline{SR} + \overline{SA} \cdot SB \cdot SR)$

COMPARISON OF UNSIGNED INTEGERS A-B

SA,SB	CS,SR = 00	01	10	11
00		A<B $a<b$	A≥B $a \ge b$	
01	A<B $a+b<2^m$	A<B $a+b \ge 2^m$		
10			A>B $a+b>2^m$	A>B $a+b \le 2^m$
11		A<B $a>b$	A≥B $a \le b$	

BORROW IS CARRY COMPLEMENTED

DOUBLE PRECISION A(B) IN AH(BH) HIGH, AL(BL) LOW

ADD: AL + BL → AL; AH + CARRY → AH; AH + BH → AH

SUB: AL - BL → AL; AH - BORROW → AH; AH - BH → AH

SUMMARY OF BRANCH CONDITIONS

PAGE REFERS TO MOTOROLA 6800 PROGRAMMING REFERENCE MANUAL

INST-PAGE-CONDITION-COMMENTS

1. BEQ A-11 $Z=1$ BRANCH ON EQUAL TO ZERO

   BNE A-20 $Z=0$ BRANCH ON NOT EQUAL TO ZERO

   BPL A-21 $N=0$ BRANCH ON PLUS

   BMI A-19 $N=1$ BRANCH ON MINUS

   BVC A-24 $V=0$ MOTOROLA DEFINES FOR ADD A+B $V = SA \cdot SB \cdot \overline{SR} + \overline{SA} \cdot \overline{SB} \cdot SR$; SUB A-B

   BVS A-25 $V=1$ $V = SA \cdot \overline{SB} \cdot \overline{SR} + \overline{SA} \cdot SB \cdot SR$.

   BCC A-9 $C=0$ ADD A+B: $C = SA \cdot SB + SB \cdot \overline{SR} + SA \cdot \overline{SR}$

   SUB A-B: $SB \rightarrow \overline{SB}$

   BCS A-10 $C=1$ TRUE CARRY $= SA \cdot \overline{SB} + \overline{SB} \cdot \overline{SR} + SA \cdot \overline{SR}$

   STATUS CARRY = " " COMPLEMENTED

   $= \overline{SA} \cdot SB + \overline{SA} \cdot SR + SB \cdot SR$ BY BOOLEAN THEOREMS.

2. CONDITIONS BASED ON SIGNED OPERATIONS

   BGE A-12 $N \oplus V = 0$ i.e. N EQUIVALENT TO V

   BLT A-18 $N \oplus V = 1$

   BGT A-13 $\overline{Z} \cdot \overline{N \oplus V} = 1$ i.e. $Z + (N \oplus V) = 0$

   BLE A-16 $Z + (N \oplus V) = 1$

3. UNSIGNED SUBTRACTOR COMPARE RESULTS

   BHI A-14 $\overline{C} \cdot \overline{Z} = \overline{C \vee Z} = 1$ i.e. $C \vee Z = 0$ { CORRECT MANUAL

   BLS A-17 $C + Z = 1$ BRANCH ON LOW OR SAME

   BHIS $\equiv$ BCC $C=0$ BRANCH ON HIGH OR SAME

   BLO $\equiv$ BCS $C=1$ BRANCH ON LOW