Source: Computer Design, Glen G. Langdon, Jr. Computeach Press, San Jose, California, 1982
Notes by JT Wunderlich PhD

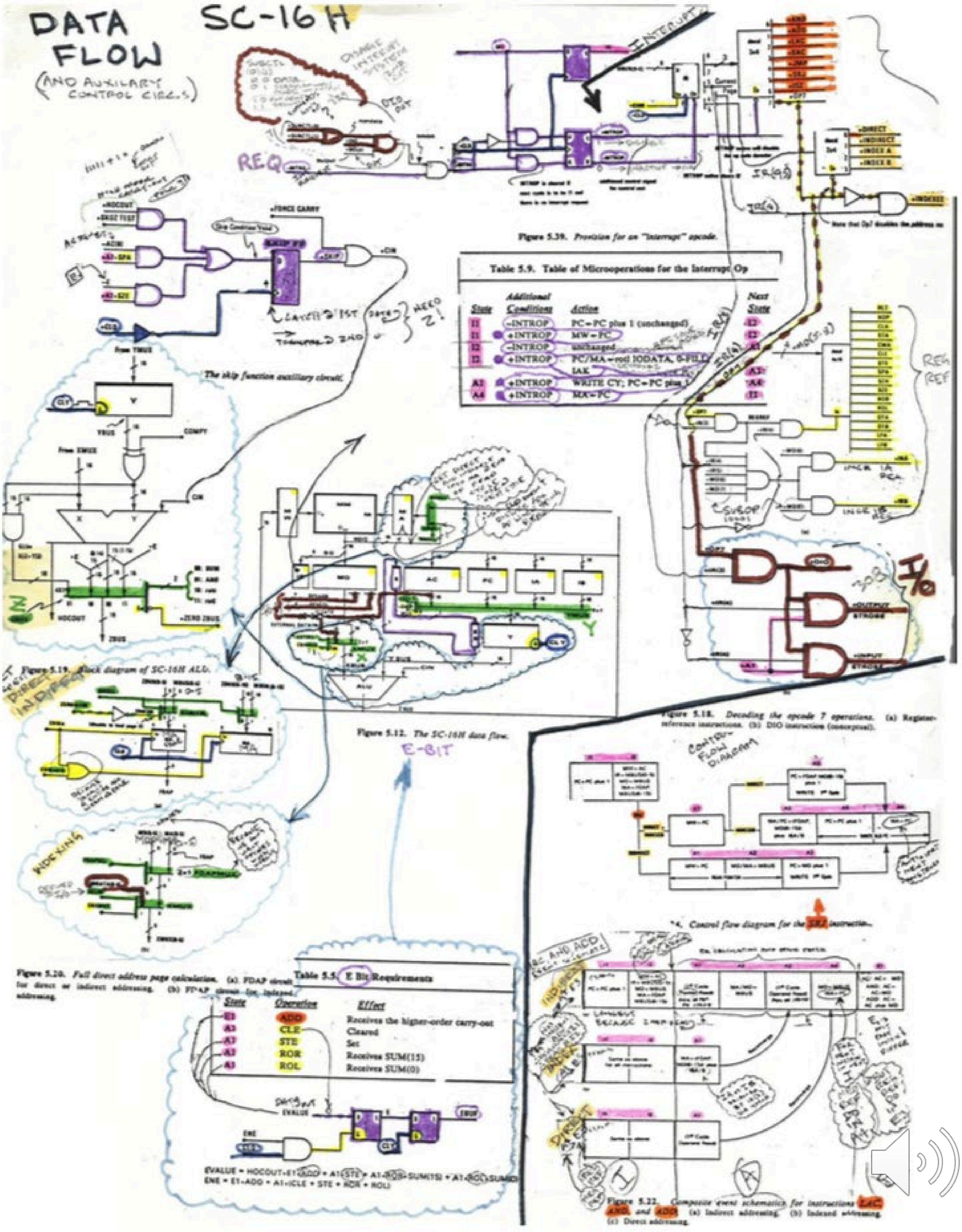


Figure 5.39. *Provision for an "Interrupt" opcode.*

Table 5.9. Table of Microoperations for the Interrupt Op

State	Additional Conditions	Action	Next State
I1	-INTROP	PC←PC plus 1 (unchanged)	I2
I1	+INTROP	MW←PC	I2
I2	-INTROP	unchanged	A1
I2	+INTROP	PC/MA←vol IODATA, 0-FILL IAK	A3
A3	+INTROP	WRITE CY; PC←PC plus 1	A4
A4	+INTROP	MA←PC	I1

The skip function auxiliary circuit.

Figure 5.19. *Block diagram of SC-16H ALU.*

Figure 5.12. *The SC-16H data flow.*

Figure 5.18. *Decoding the opcode 7 operations.* (a) Register-reference instructions. (b) DIO instruction (conceptual).

Figure 5.20. *Full direct address page calculation.* (a) FDAP circuit for direct or indirect addressing. (b) FDAP circuit for indexed addressing.

Table 5.5. E Bit Requirements

State	Operation	Effect
E1	ADD	Receives the higher-order carry-out
A1	CLE	Cleared
A1	STE	Set
A1	ROR	Receives SUM(15)
A1	ROL	Receives SUM(0)

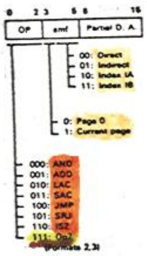
EVALUE = HOCOUT·E1·ADD + A1·STE + A1·ROR·SUM(15) + A1·ROL·SUM(0)
ENE = E1·ADD + A1·(CLE + STE + ROR + ROL)

4. *Control flow diagram for the SRJ instruction.*

Figure 5.22. *Composite event schematics for instructions LAC, AND, and ADD.* (a) Indirect addressing. (b) Indexed addressing. (c) Direct addressing.

SC-16H

CONTROL



① MEMORY REFERENCE INSTRUCTIONS

Figure 6.7. *SC-16 instruction format 1, memory-reference instructions.*

AND	- op 0:	AC←AC and M[ea]; E bit is left unchanged.
ADD	- op 1:	AC←AC plus M[ea]; E bit receives the carry-out
LAC	- op 2:	AC←M[ea]; E bit unchanged.
SAC	- op 3:	M[ea]←AC; E bit unchanged.
JMP	- op 4:	PC←ea; E bit unchanged.
SRJ	- op 5:	M[ea]←PC; PC←ea plus 1; E bit unchanged.
ISZ	- op 6:	M[ea]←M[ea] plus 1; E bit unchanged; 0 = M[ea] (after the increment): PC←PC plus 1.

Table 5.3. ALU Functions Implied by the Opcodes

Opcode	*Function*
ADD	Binary addition
AND	Logical (bit-by-bit) AND
ISZ, SZA	INC One-up and/or Zero-detect
ROR	Rotate right
ROL	Rotate left
CLA	Place zeros on ALU out
CMA, STA	Complement Y input
SPA	Detect sign AC(0)
CLE, STE, SZE	Manipulate and test E bit

WRITE ALL CONTROL SIGNALS FROM DEPENDENT

WE ASSUME PATH EXISTS

TEST FOR INTERUPT

	Conditions (p-term)	Microoperations (RTN)	PLA p-term	Next State
	I1	PC←PC plus 1	1	I2
	I2	MW←AC; IR←MBUS(0-5); MA(6-15)←MBUS(6-15); MD←MBUS	2	A1
	I2·$\overline{\text{MBUS(3)'}}$	MA(0-5)←0	3	A1
	A1·(INDIRECT+SRJ·INDEXED)	(p-term for Next State only)		A2
5.	A1·INDEXED·SRJ'·JMP'	MA←FDAP,MD(6-15) plus I5A/B ITOY	4	A3
6.	A1·DIRECT·SRJ'·JMP'	(p-term for Next State only)		A4
7.	A1·DIRECT·SRJ	(p-term for Next State only)		A3
8.	A1·DIRECT·SAC	WRITECY	5	A4
9.	A1·DIRECT·JMP	PC←FDAP,MD(6-15)	6	I1
10.	A1·SRJ	MW←PC	7	lines 4, 7
11.	A1·INDEXED·JMP	PC/MA←FDAP,MD(6-15) plus I5A/B ITOY	8	I1
12.	A2·INDEXED'	MA/MD←MBUS	9	A3
13.	A2·INDEXED	PC/MA←FDAP,MD(6-15) plus I5A/B ITOY	10	A3
14.	A3·JMP'	(p-term for Next State Only)		A4
15.	A3·(SAC+SRJ)	WRITECY	11, 12	A4
16.	A3·JMP	PC←MD	13	I1
17.	A3·SRJ·DIRECT	PC←FDAP,MD(6-15) plus 1	14	A4
18.	A3·SRJ·INDIRECT	PC←MD plus 1	15	A4
19.	A3·SRJ·INDEXED	PC←PC plus 1	16	A4
20.	A4·SRJ'·SAC'	MD←MBUS	17	E1
21.	A4·ISZ	MA←PC	18	lines 20, 22
22.	A4·(SAC+SRJ)	(p-terms for Next State only)		I1
23.	E1·LAC	AC←MD	19	I1
24.	E1·AND	AC←AC·MD	20	I1
25.	E1·ADD	AC←AC plus MD	21	I1
26.	E1·ISZ	MW←MD plus 1	22	E2
27.	E2·ISZ	WRITECY; ~~No dest←MD plus 1~~; SKOZTEST	23	E3
28.	E3·ISZ	PC/MA←PC (add SKIP FF via CIN)	24	I1

Table of microoperations for memory-reference

BECAUSE

FF ADDR MODEL SEQ. MACHINE PRESENT STATE

ALL THIS TRUE ONLY FOR

NOT TALK BEHAVIOR UPPER...

Table 5.4. Table of Microoperations for LAC, AND, and ADD

MAJOR *Control State*	*Additional Conditions* = f(OPCODE)	CONDITION *Microoperation*	*Next State*
I1	WAITING (READING)	PC←PC plus 1	I2

I2		MW←AC; IR←MBUS(0-5);
		MD←MBUS;
		MA←FDAP, MBUS(6-15)
			A1

A1	INDIRECT· (LAC+AND+ADD)		A2
A1	INDEXED· (LAC+AND+ADD)	MA←(FDAP, MD(6-15)) plus I5A/B	A3
A1	DIRECT· (LAC+AND+ADD)		A4
A2	INDIRECT· (LAC+AND+ADD)	MA/MD←MBUS	A3
A3	(LAC+AND+ADD)		A4
A4	LAC+AND+ADD	MD←MBUS; MA←PC	E1
E1	LAC	AC←MD	I1
E1	AND	AC←AC·MD	I1
E1	ADD	AC←AC plus MD	I1

"SAVE IS GREAT" ... ACTION ... "THIS CANCELS"

REGISTER REF. INSTRUC. ②

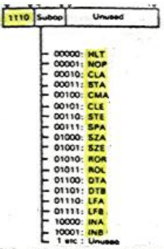


Table 5.2. Register-Reference Instructions

Name	*Subop*	*Description*
HLT	0	Halt the computer. Wait for console interrupt to restart the machine.
NOP	1	No operation. Continue with next instruction.
CLA	2	Clear the AC (AC←0).
STA	3	Set AC to all 1's (AC←1).
CMA	4	Complement the AC (AC←AC').
CLE	5	Clear the E bit (CLRE).
STE	6	Set E bit to 1 (SETE).
SPA	7	Skip next instruction if AC is positive (AC(0)=0: PC←PC plus 1).
SZA	8	Skip next instruction if AC is zero (AC=0: PC←PC plus 1).
SZE	9	Skip next instruction if E bit is 0 (E=0: PC←PC plus 1).
ROR	10	Rotate AC right, E bit provides the fill and accepts the spill (rotr (E,AC)).
ROL	11	Rotate AC left, E bit provides the fill and accepts the spill (rotl (E,AC)).
DTA	12	Deposit contents of AC into index register IA (IA←AC).
DTB	13	Deposit contents of AC into index register IB (IB←AC).
LFA	14	Load the AC from the IA register (AC←IA).
LFB	15	Load the AC from the IB register (AC←IB).
INA	16	Increment the IA register by 1 (IA←IA plus 1).
INB	17	Increment the IB register by 1 (IB←IB plus 1).

304 321

Condition	RTN Statement (MICRO-P. DECOMPOSE INTO)	Corresponding Control Signals (NOT RTN)	Next State (E3 BECAUSE)
A1·HLT	Clear RUN flip-flop	HALT (Fig. 5.45)	E3
A1·NOP			E3
A1·CLA	AC←0	ENAC; DISYMUX	E3
A1·STA	AC←1...1	ENAC; DISYMUX; COMPY	E3
A1·CMA	AC←AC'	ENAC; COMPY	E3
A1·CLE	E←0	CLEAR E	E3
A1·STE	E←1	SET E	E3
A1·SPA	SKIP←AC(0)'	Auxiliary Circuit (Fig. 5.26)	E3
A1·SZA	No dest←AC' plus 1	SKOZTEST; FORCE CARRY; COMPY;	E3
A1·SZE	SKIP←E'	Auxiliary Circuit (Fig. 5.26)	E3
A1·ROR	AC←rotr AC	ENAC; ROTATER	E3
A1·ROL	AC←rotl AC	ENAC; ROTATEL	E3
A1·DTA	IA←AC	ENIA	E3
A1·DTB	IB←AC	ENIB	E3
A1·DFA	AC←IA	ENAC; IA2Y (Y mux select)	E3
A1·DFB	AC←IB	ENAC; IB2Y (Y mux select)	E3
A1·INA	IA←IA plus 1	ENIA; IA2Y; FORCE CARRY	E3
A1·INB	IB←IB plus 1	ENIB; IB2Y; FORCE CARRY	E3

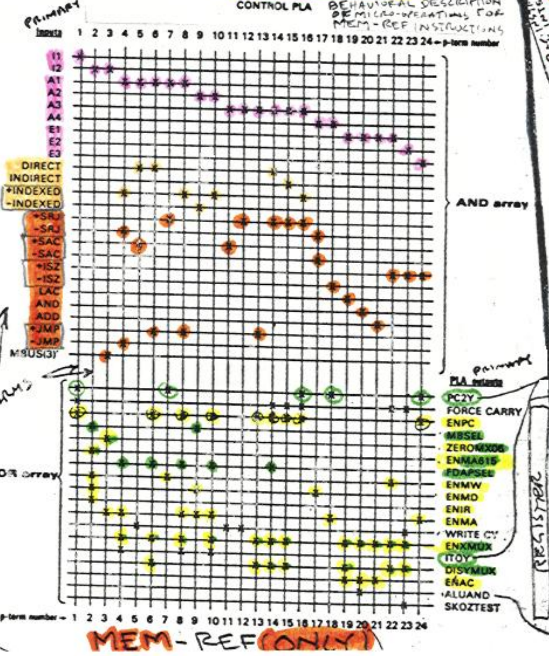
PC→4A

E3

Table 5.7. Decomposing RTN Statements to Control Points

RTN Statement Level	IMPLIED! *Control Point Activation Level*
PC←PC plus 1	PC2Y; FORCE CARRY; ENPC
PC/MA←MD plus I5A/B	ENXMUX; ITOY; ENMA; ENPC
AC←AC·MD	ENXMUX; ALUAND; ENAC

CONTROL PLA BEHAVIORAL DESCRIPTION OF MICRO-OPERATIONS FOR MEM-REF INSTRUCTIONS



MEM-REF (ONLY)

I/O ③ INSTRU

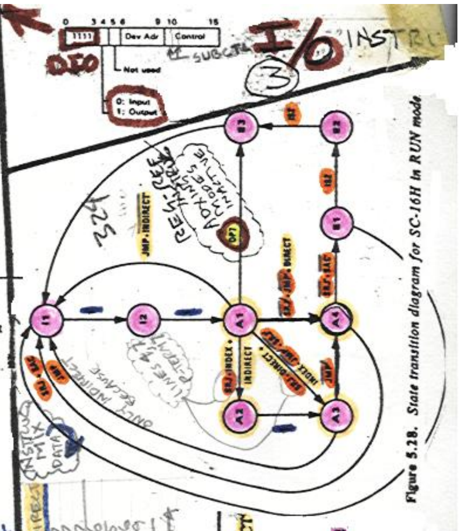


Figure 5.28. State transition diagram for SC-16H in RUN mode

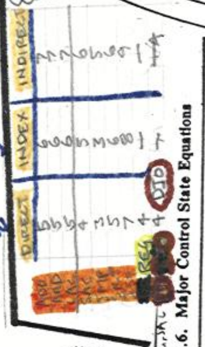


Table 5.6. Major Control State Equations

D11	=	RUNMODE (A4(SRJ+SAC) + A1·JMP·INDIRECT + A3·JMP + E1·ISZ' + E3) +RUNCY
D12	=	I1
DA1	=	I2
DA2	=	A1 (SRJ·INDEX + INDIRECT)
DA3	=	A1·DIRECT·SRJ + A1·INDEX·JMP'·SRJ' + A2
DA4	=	A1·SRJ'·JMP'·DIRECT + A3·JMP'
DE1	=	A4·SRJ'·SAC'·DIRECT
DE2	=	E1·ISZ
DE3	=	A1·OP7 + E2·ISZ

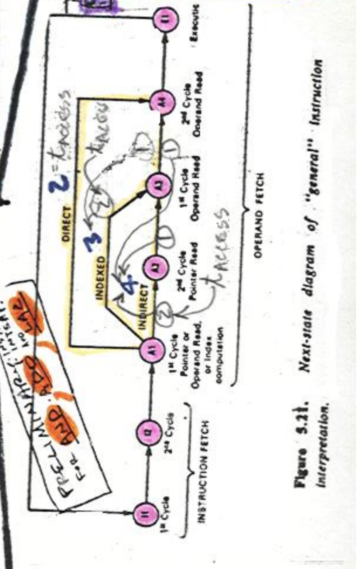


Figure 5.21. Next-state diagram of "general" instruction interpretation.

YSEL A	YSEL B	Source Selec
0	0	AC
0	1	IA
1	0	PC
1	1	IB

YSEL A = PC2Y + ITOY·IR(5) + IB2Y

YSEL B = ITOY + IA2Y + IB2Y

IR(5)'

ZSEL A	ZSEL B	Source Selec
0	0	SUM
0	1	ALUAN
1	0	rotr
1	1	rotl

ZSEL A = ROTATER + ROTATEL

ZSEL B = ALUAND + ROTATEL

TIMING

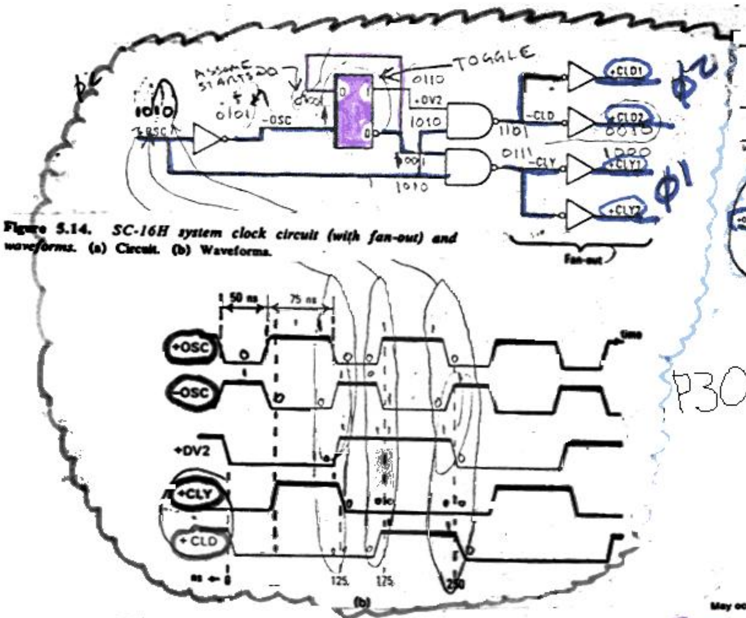


Figure 5.14. *SC-16H system clock circuit (with fan-out) and waveforms.* (a) Circuit. (b) Waveforms.

P304

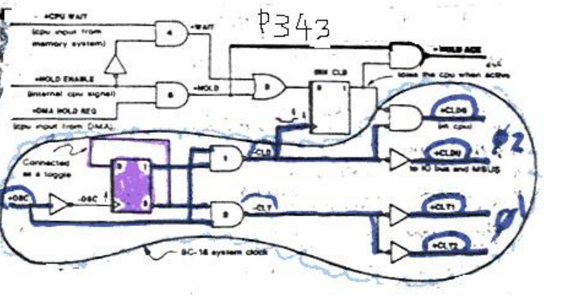
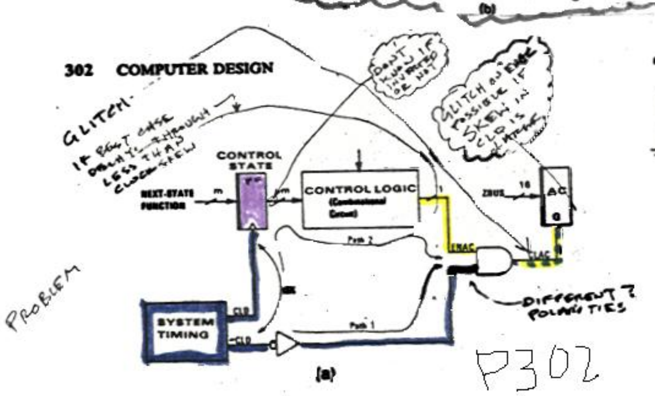
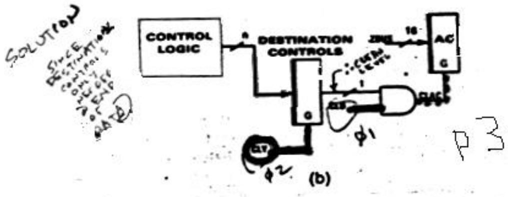


Figure 5.46. *CPU idling circuit.*


GLITCH
IF BEST CASE DELAY THROUGH LESS THAN CLOCK SKEW
DON'T KNOW IF INVERTED OR NOT
GLITCH ON EDGE POSSIBLE IF SKEW IN CLD IS GREATER
CONTROL STATE
CONTROL LOGIC (Combinational Circuit)
NEXT-STATE FUNCTION
ZBUS
16
AC
G
ENAC
CLAC
Path 2
Path 1
SYSTEM TIMING
CLD
-CLD
DIFFERENT ? POLARITIES
PROBLEM
(a)

302 COMPUTER DESIGN

P302



P302

Figure 5.13. *Timing for register clocking.* (a) Illustration of a race condition upstream of CLAC. (b) Destination control register removes race condition.

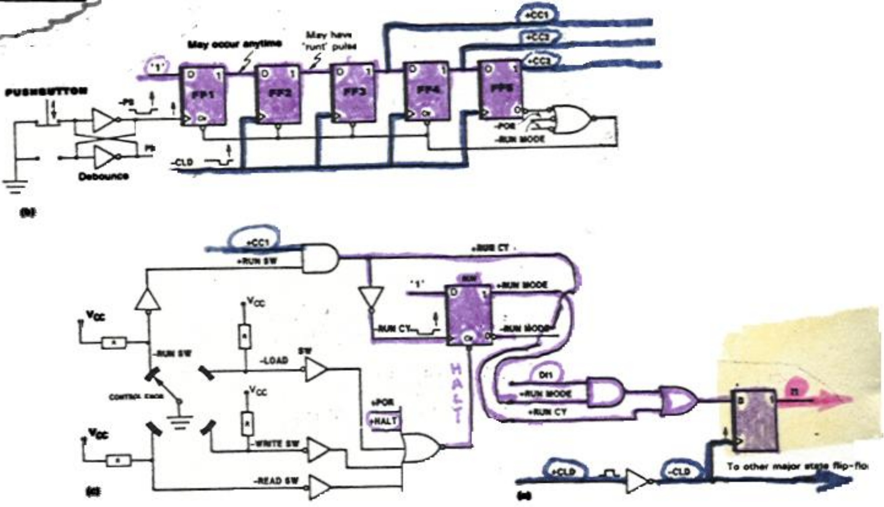
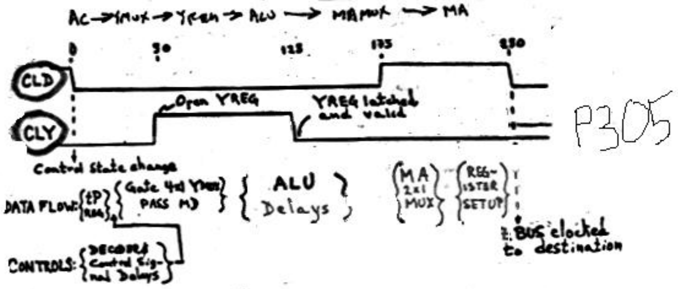


Figure 5.45. *Auxiliary control circuits involving console operations.* (a) Suspension of state I1, and restart. (b) Generation of console operation control states. (c) Control knob and RUN operation.

P344


AC→YMUX→YREG→ALU→MAMUX→MA
0
50
125
175
250
CLD
CLY
Open YREG
YREG latched and valid
Control state change
DATA FLOW: {tP (ROM)} {Gate 4:1 Yreg PASS M} {ALU Delays} {MA 2x1 MUX} {REGISTER SETUP}
BUS clocked to destination
CONTROLS: {DECODERS Control Signal Delays}

P305

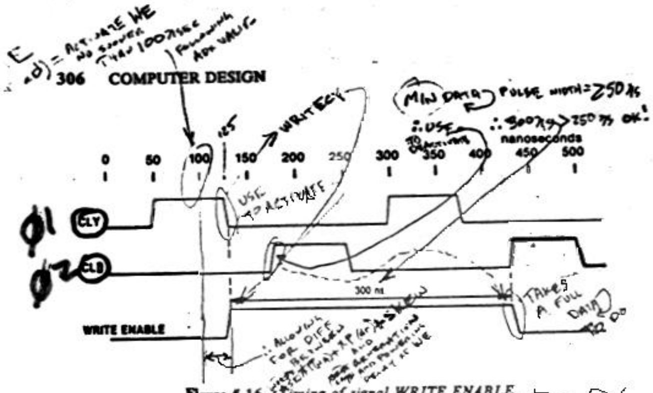
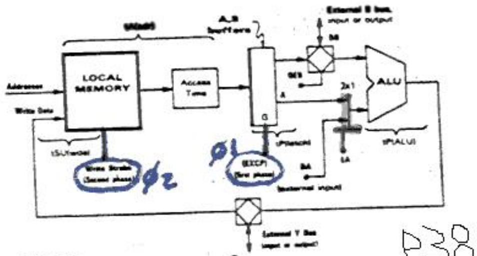
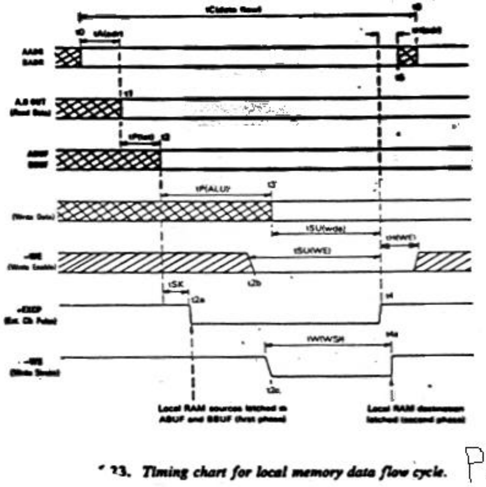


Figure 5.16. *Timing of signal WRITE ENABLE.*

P306



P389



23. Timing chart for local memory data flow cycle.

P390

SG16-H

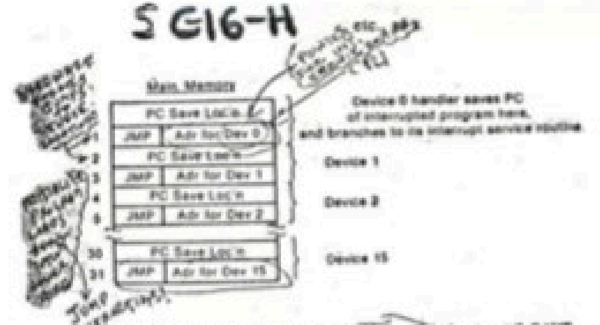
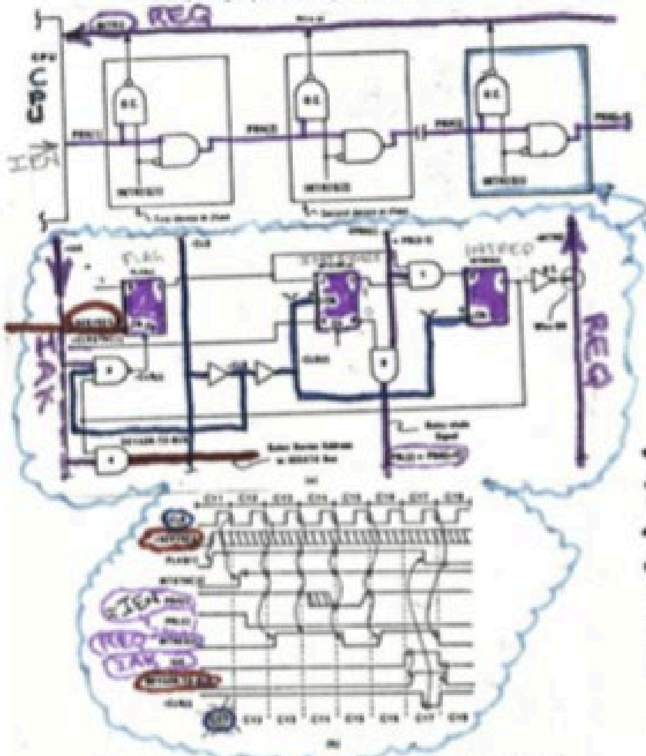


Figure 5.36. Map of main memory (low core) showing PC SAVE areas, and jumps to interrupt service routines for the devices.



INTERRUPT

Figure 5.38. Logic and timing of device interrupt logic. (a) Logic schematic. (b) Timing chart.

MA←PC	PC/MA←(0) IODATA,0 IAK	PC←PC plus 1 WRITE C1	MA←PC

Figure 5.40. Instruction event schematic for interrupt acceptance.

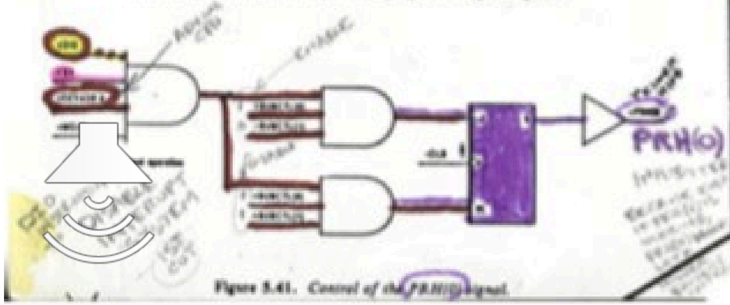


Figure 5.41. Control of the PRH(0) signal.

DMA

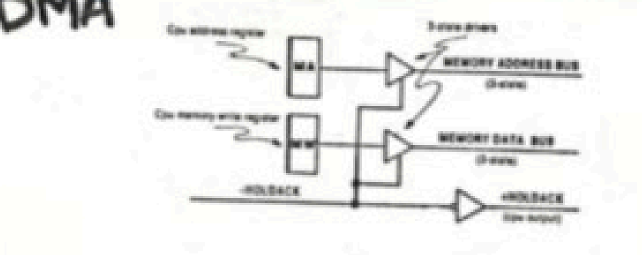


Figure 5.43. CPU releasing the memory bus.

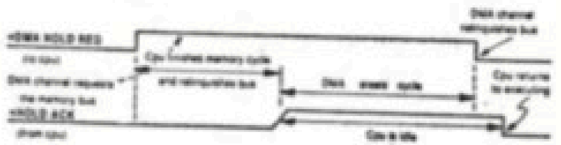
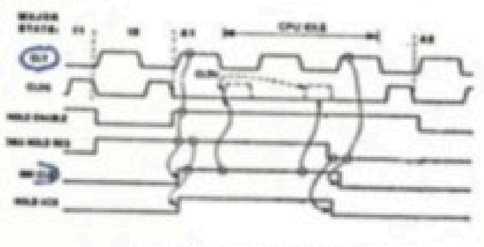


Figure 5.44. Timing sequence which coordinates a DMA "cycle-steal".



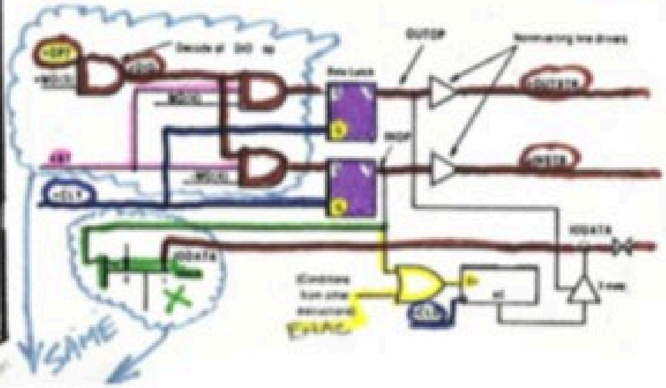
Synchronization of a DMA memory cycle

Table 5.8. Execution of the DIO Instruction	

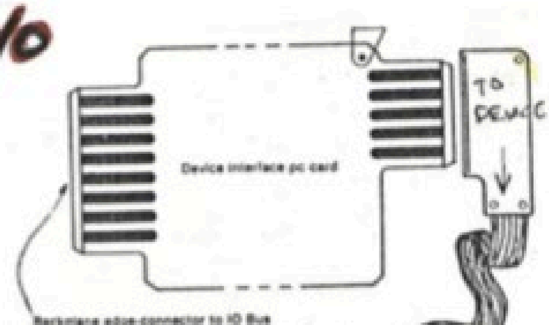
DIO·A1	SUBCTL = MD(14-15);
	DEVSEL = MD(6-9)

DIO·A1·MD(4)	OUTSTR = 1
DIO·A1·MD(4)'	INSTR = 1
DIO·A1·MD(4)'	AC ← IODATA

NEXT STATE

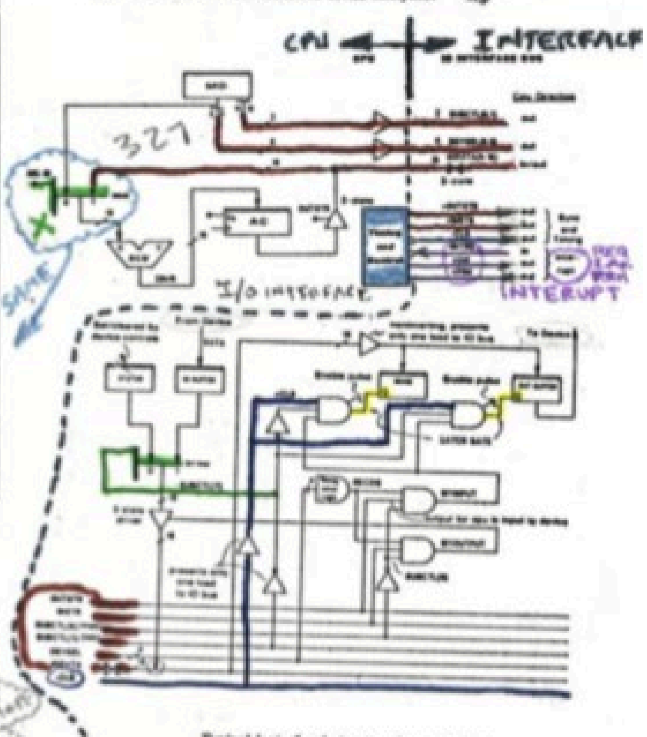


I/O

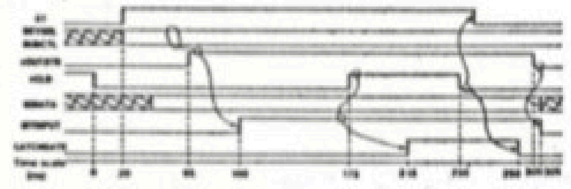


Backplane edge-connector to IO Bus

Figure 5.33. IO bus connection to device interface.



Typical logic for device interface to IO bus.



Timing chart for output instruction.