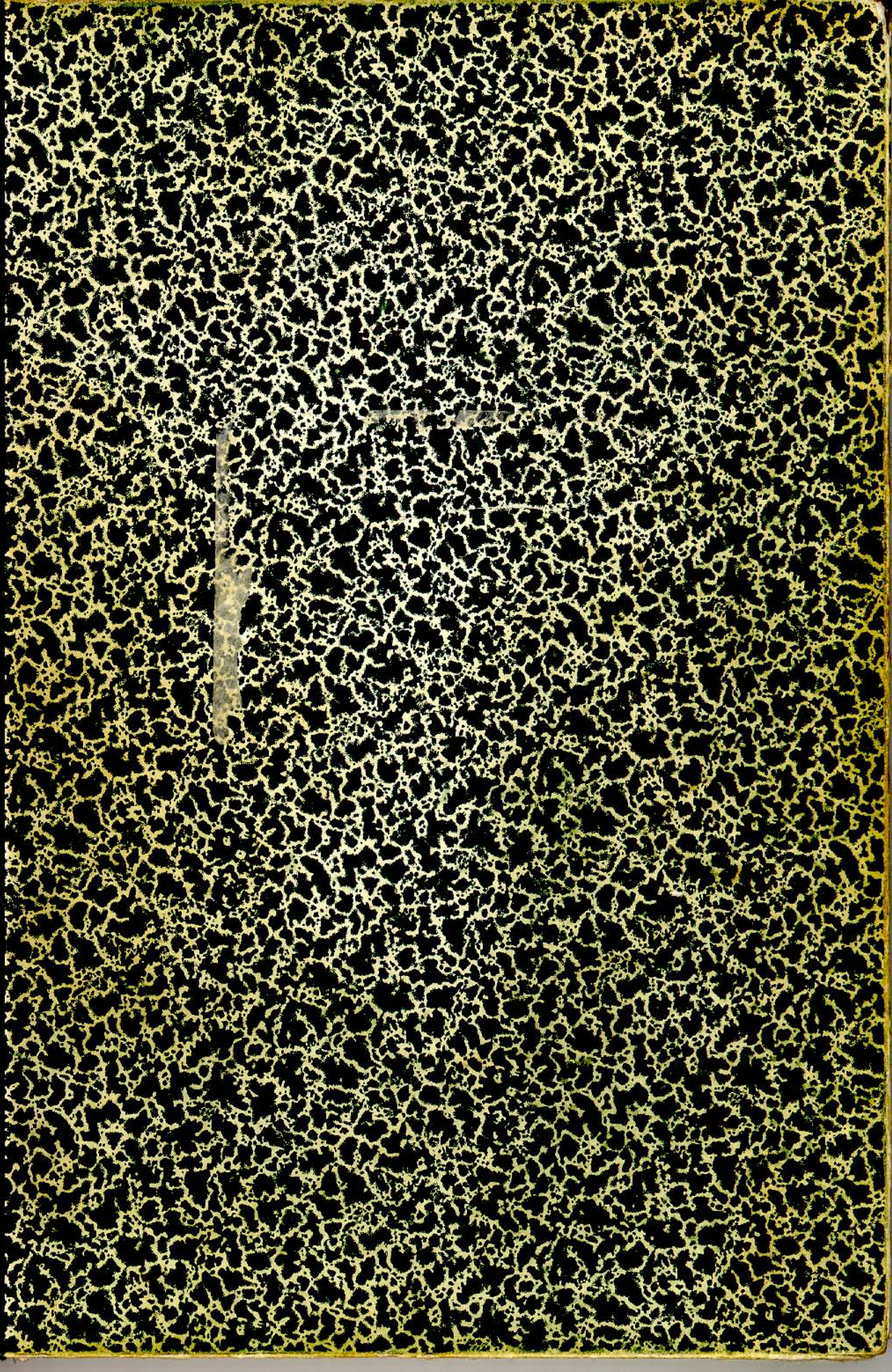


COMPUTER DESIGN (1)

TABLE OF CONTENTS
(3 March - 7 June 1976)

7 June 1976
BFSD

PP

32	N, V, C, Z Bit Selector Logic
33	Shift Rotate Control Logic
34	Word/Byte Condition Code Select Logic Byte Enable Logic
35	Carry IN Logic for MPY
36-37	A & B Port Address Multiplexers
38	Refined Dest/Source Addressing Code
* 39-40	Second trial Microcontroller Logic
* 41-51	First IR Decoding Logic
* 52	Optimal SPL Logic (11/45)
* 53-54	First I/O Control Sequencer
55-56	Address Register & BUS Drivers
57	Modification in Interrupt Handling
* 58	INTR Programming
59-62	First Trial NPR & BRANCH Request Logic
* 63-65	First Trial Complete I/O BUS Control Logic
66	Currently Designated Microcode Lines
67	Preliminary BUS Address Decoder for PSW, SLR, Console Switches
68-69	Revising PSW Logic
* 70-72	Redesign of PR Logic
* 73	Priority Logic
* 74	clear Selected Priority Logic

pp	
75	Changes in BUS I/O described
76	Notes on the Lookahead DATI IR
77-79	IR Decoding data
80-81	Writable Control Stores / Control Register listing
82-85	IR Decoding Logic
86-87	Redesign of the Microcontroller Logic (AM2909)
88	Redesign of DReg / BUS I/O Logic
89	Redesign of CPU Clock
90	Redesign of Counter Logic
91-94	Redesign of I/O BUS Control Logic
95	CPU Same Signals (Timing Diagrams)
96-97	EIS for MPY, DIV, ASH, ASHC
98-99	Redesign of High Order Priority Logic Clean Current Priority Logic
100-101	Priority Clock Trap Group Priority Encoder
102	Program Priority Encoder BR Encode Address Multiplexer

* indicate deleted items

7 June 1976
ARD

Summary of Condition Code Settings for PDP-11 instructions

	N (=1 if)	Z (=1 if)	V	C
CLR (B)	∅	1	∅	∅
COM (B)	<∅	=∅	∅	1
INC (B)	<∅	=∅	overflow	——
DEC (B)	<∅	=∅	overflow	——
NEG (B)	<∅	=∅	overflow	$\overline{Z}$
ADC (B)	<∅	=∅	overflow	carry
SBC (B)	<∅	=∅	overflow	carry
TST (B)	<∅	=∅	∅	∅
ROR (B)	<∅	=∅	N⊻C	C from Rotate
ROL (B)	<∅	=∅	N⊻C	C from Rotate
ASR (B)	<∅	=∅	N⊻C	C from Shift
ASL (B)	<∅	=∅	N⊻C	C from Shift
SWAB	<∅ Set from Lower Byte (after SWAB)	=∅	∅	∅
SXT	<∅	=∅	∅	——

3 March 1976
ARB

Condition Codes

	N	Z	V	C
MOV (B)	$< \emptyset$	$= \emptyset$	$\emptyset$	——
CMP (B)	$< \emptyset$	$= \emptyset$	overflow	carry
ADD	$< \emptyset$	$= \emptyset$	overflow	carry
SUB	$< \emptyset$	$= \emptyset$	overflow	carry
BIT (B)	$< \emptyset$	$= \emptyset$	$\emptyset$	——
BIC (B)	$< \emptyset$	$= \emptyset$	$\emptyset$	——
BIS (B)	$< \emptyset$	$= \emptyset$	$\emptyset$	——
XOR	$< \emptyset$	$= \emptyset$	$\emptyset$	——
MFPI	$< \emptyset$	$= \emptyset$	$\emptyset$	——
MTPI	$< \emptyset$	$= \emptyset$	$\emptyset$	——
MUL	$< \emptyset$	$= \emptyset$	$\emptyset$	carry
DIV	$< \emptyset$	$= \emptyset$	1 if Source $= \emptyset$ 1 if overflow	1 if Source $= \emptyset$
ASH	$< \emptyset$	$= \emptyset$	$N \neq N'$ ie if sign changed	loaded
ASHC	$< \emptyset$	$= \emptyset$	$N \neq N'$ ie if sign changed	loaded

Condition Codes

	N	Z	V	C
FADD	$<\emptyset$	$=\emptyset$	$\emptyset$	$\emptyset$
FSUB	$<\emptyset$	$=\emptyset$	$\emptyset$	$\emptyset$
FMUL	$<\emptyset$	$=\emptyset$	$\emptyset$	$\emptyset$
FDIV	$<\emptyset$	$=\emptyset$	$\emptyset$	$\emptyset$

for errors TRAP Through Location 244

the V = 1 ; if an error

N = 1 ; if underflow or divide-by-zero

C = 1 ; if divide by Zero

Z = $\emptyset$

3 March 1826

ABB

Additional Code -

[17 words] Preliminary Destination Mode Microcode

Destination Address left in AD

Destination Data left in DI

Stack check Register EXR6 Loaded for all modes - 1, 2, 4 & 6

FETCH = DATIP/or DATI (special) ; RX from Destination Operand OR NO Access on Certain Instructions (NDA) with Enable for DATI (SE)

Mode 1 WI/O[L] RX → AD ; BR to [END] ; FETCH ; Special Enable

Mode 2 WI/O[L] RX → AD ; Rx+1 → RX ; Fetch ; Special Enable
Rx+1 → Rx ; (Byte conditioned or R6/R7)
BR to [END]

Mode 3 WI/O[L] RX → AD ; Rx+1 → R x ; DATI
Rx+1 → Rx
[OUT 1] WAIT FOR I/O [H] ; DI → AD ; BR to [IRCODE] ; FETCH (SE)

Mode 4 Rx-1 → Rx (Byte Conditioned or R6/R7)
[OUT 0] WI/O[L] Rx-1 → AD ; BR [END] ; FETCH (SE)

Mode 5 Rx-1 → Rx
WI/O[L] RX-1 → RX & AD → ; DATI
; BR to [out 1]

Mode 6 WI/O[L] R7 → AD ; R7+1 → R7 ; DATI
R7+1 → R7
WAIT FOR I/O [H] ; DI+Rx → AD ; BR to [END] ; FETCH (SE)

Mode 7 WI/O[L] R7 → AD ; R7 + 1 → R7 ; DATI
R7 → 1 → R7
WAIT FOR I/O [H] ; DI+RX → AD ; BR [OUT 1] ; DATI

[END] R6 - STL → SOR ; BR [IRCODE]

3 March 1976
AMD

WI/O — Wait for I/O

Destination Instruction times in Cycles + Memory Access

(computed for 200NS Microcycle + 600NS Access)
(800 NS)

	Micro + Mem	xfer + Store
Mode 1	1 1	.8 (1.0) .3 .325
Mode 2	1 1	.8 (1.0) .3
Mode 3	2 2	1.6 (2.0) .3
Mode 4	2 1	1.0 (1.2) .3
Mode 5	3 2	1.8 (2.2) .3
Mode 6	2 2	1.6 (2.0) .3
Mode 7	3 3	2.4 (3.0) .3

for NDA's subtract 1 Memory Access Cycle
and add 1 Micro cycle for Modes 1, 2, 4 & 6

On Byte DATO's BUS control
positions byte properly for write sequence

[18 words] Preliminary Source Mode Microcode

Program Counter left in AD

Source Data left in TØ ; All fetches are DATI

Stack check Register EXR6 loaded after Modes 1, 2, 4 & 6

Rx from Source operand

Mode 1 — WI/O [L]; Rx → AD ; BR [END] ; DATI (BE) ← FETCH

Mode 2 — WI/O [L]; Rx → AD ; Rx+1 → RX ; DATI (BE)
Rx+1 → RX (conditional on Byte or R6/R7)
; BR [END]

Mode 3 — WI/O [L]; Rx → AD ; Rx+1 → RX ; DATI
Rx+1 → RX
[OUT1] WI/O [H] ; DI → AD ; BR [END1] ; DATI (BE)

Mode 4 — Rx-1 → RX (conditional on Byte or R6/R7)
WI/O [L]; Rx-1 → Rx + AD; BR [END] ; DATI (BE)

Mode 5 — Rx-1 → RX
WI/O [L] ; Rx-1 → Rx+AD; ; DATI
; BR [OUT 1]

Mode 6 — WI/O [L] R7 → AD; R7+1 → R7 ; DATI
R7+1 → R7
WI/O [H]; DI + Rx → AD; BR [END]; DATI (BE)

Mode 7 — WI/O [L] R7 → A; R7+1 → R7 ; DATI
R7+1 → R7
WI/O [H]; DI + Rx → AD; BR [OUT1] ; DATI

[END] R6 − SLR → SoR
[END1] WI/O [H] ; R7 → AD; DI → TØ; BR ERCODE]

3 March 76
ARD

Source Instruction times in Cycles + Memory Access [23

(computed for 200 ns cycle and 600 ns Access

	Micro	+	Mem	μsec
Mode 1	2		1	1.0
Mode 2	2		1	1.0
Mode 3	3		2	1.8
Mode 4	3		1	1.2
Mode 5	4		2	2.0
Mode 6	3		2	1.8
Mode 7	4		3	2.6

for Byte Mode Source - Addressed Byte is loaded into <07:00>; ~~its sign extended~~ ~~loaded into <15:08> automatically~~ 67 me 76

[23 words] Single Operand Microcode (DD)

Assumes Destination Code has been executed with Address in AD and Data coming into DI for Modes 1-7

SCC – Set condition Codes

CLR (B) NDA — WI/O[L]; ∅ → DO; SCC; BR[BEGIN]
Byte Enable (BE); DATO (B)

COM (B) — WI/O[H]; $\overline{DI}$ → DO; SCC; BE; BR [BEGIN]; DATO (B)

INC (B) — WI/O[H]; DI+1 → DO; SCC; BE; BR [BEGIN]; DATO (B)

DEC (B) — WI/O[H]; DI-1 → DO; SCC; BE; BR[BEGIN]; DATO (B)

NEG (B) — WI/O[H]; $\overline{DI}$+1 → DO; SCC; BE; BR[BEGIN]; DATO (B)

TST (B) (special) — WI/O [H]; DI+∅ → DO; SCC; BE; BR [BEGIN]
R7 → AD; DATI IR

SWAB — WI/O [H]; DI → SWAB +0 → DO; SCC; BR [BEGIN]; DATO

ADC (B) — WI/O [H]; DI+C → DO; SCC; BE; BR [BEGIN]; DATO (B)

SBC (B) — WI/O [H]; DI-C → DO; SCC; BE; BR [BEGIN]; DATO (B)

SXT NDA — WI/O [L]; #O $\begin{matrix}-1 \text{ (if } N=1)\\ +0 \text{ (if } N=0)\end{matrix}$ → DO; SCC; BR [BEGIN]; DATO

XOR — WI/O [H]; DI ∀ RX → DO; SCC; BR [BEGIN]; DATO

JMP (special) — WI/O[H]; DI → R7 + AD; BR [BEGIN x]; DATI IR

JSR (special) — R6-1 → R6
R6-1 → R6 + EXR6
WI/O [H]; DI → TØ; R6 → AD
RX → DO; DATO
R7 → RX
TØ → R7; BR BEGIN

3 March 1976
ARD

Shift Rotate Group for Modes 1-7

ROR(B) WI/O[H] ; DI → ROR → TØ ; SC ; BR [OUT] ; BE

ROR (B) WI/O [H] ; DI → ROL → TØ ; SC ; BR [OUT] ; BE

ASR (B) WI/O [H] ; DI → ASR → TØ ; SC ; BR [OUT] ; BE

ASL (B) WI/O [H] ; DI → ASL → TØ ; SC ; BR [OUT] ; BE

[OUT] TØ+Ø → DO ; SCC ; BR [BEGIN] ; DE DATO (B)

Execution times Single Operand (OD)

Modes 1-7

Micro cycles of 200 NS Memory of 300 ns on DATO, 600 ns on DATI

	Micro	DATO's	DATI's	Time
CLR (B)	0	1	0	.3
COM (B)	1	1	0	.5
INC (B)	1	1	0	.5
DEC (B)	1	1	0	.5
NEG (B)	1	1	0	.5
TST (B)	1	0	0	.2
SWAB	1	1	0	.5
ADC (B)	1	1	0	.5
SBC (B)	1	1	0	.5
SXT	1	1	0	.5
XOR	1	1	0	.5
JMP	1	0	0	.2
JSR	2	~~1~~ 0 (See Micro Code)	0	.4
ROR (B)	2	1	0	.7
ROL (B)	2	1	0	.7
ASR (B)	2	1	0	.7
ASL (B)	2	1	0	.7

3 March 76
ARD

[19 words] Single Operand Microcode Mode 0

Assumes Next PC is in AD (LC = Load Bit)

CLR(B) BE; 0 → RX; SCC; BR [BEGIN]√lookahead

COM(B) BE; $\overline{RX}$ → RX; SCC; BR [BEGIN]√lookahead

INC (B) BE; RX+1 → RX; SCC; BR [BEGIN]√lookahead

DEC (B) BE; RX−1 → RX; SCC; BR [BEGIN]√lookahead

NEG(B) BE; $\overline{RX}$+1 → RX; SCC; BR [BEGIN]√lookahead

TST (B) BE; RX → RX; SCC; BR [BEGIN]+1

SWAB RX → DO;
DI → SWAB+0 → RX; SCC; BR [BEGIN]√lookahead

ADC (B) BE; RX+C → RX; SCC; BR [BEGIN]√lookahead

SBC (B) BE; RX−C → RX; SCC; BR [BEGIN]√lookahead

SXT 0 $\begin{matrix}-1\ (N=0)\\ +0\ (N=1)\end{matrix}$ → RX; SCC; BR [BEGIN]√lookahead

XOR RS ∀ RD → RD; SCC; BR [BEGIN]√lookahead

JMP ILLEGAL INSTRUCTION —

JSR ILLEGAL INSTRUCTION —

ROR (B) RX → ROR → RX; LC
[OUT] RX+0 → RX; SCC; BR [BEGIN]√lookahead

ROL (B) RX → ROL → RX; LC; BR [OUT]

ASR (B) RX → ASR → RX; LC; BR [OUT]

ASL (B) RX → ASL → RX; LC; BR [OUT]

lookahead = DMode ϕ [H] ∧ $\overline{R7\,[H]}$

Times = .2 μsec (microcycle)

DATIIR includes interrupt enable

3 March 76
ARD

[3 words] Preliminary Begin Microcode.

[BEGIN] WI/O[L] R7 → AD ; DATI IR
[BEGINX] WI/O[H] conditional Branch on Interrupts or BUS Errors

R7+#2 → R7+AD ; DECODE IR ; BR [IRCODE]

~~Interrupt Pending — DATI IR Aborted~~

Execution time [BEGIN] 1.0 μsec
[BEGINX] .8 μsec
@ .2 μsec Microcycle .6 μsec DATI IR

Lookahead

IDATI IR BR
[Begin] WI/O [L] ; R7 → AD ; INTCHH I/O ; BR
[Begin]X or [Begin]V1 WI/O [L] ; R7+#2 → R7+AD ; INTCHH I/O
WI/O [H] BR [ICODE] ; Conditional Lookahead DATI IR
15Jan77 IR<07:00> SXT *2 → T0

(1 June 1976)
ARD

3 March 76
ARD

Lookahead for Double OPs

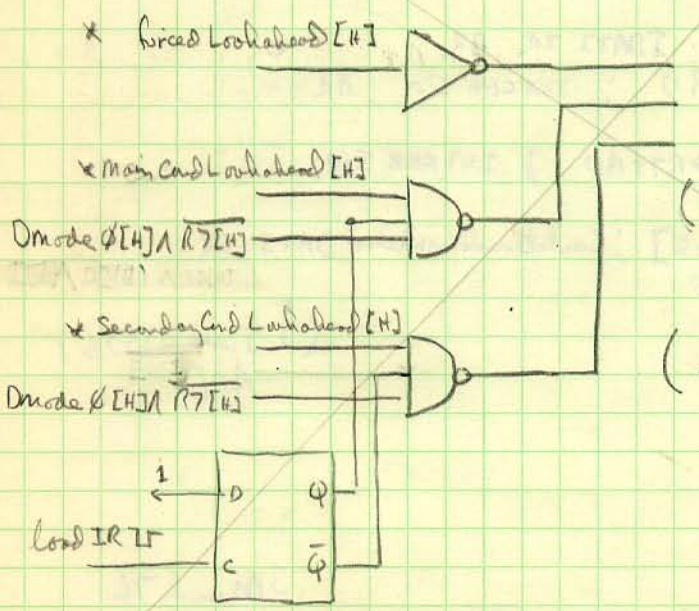
SS 1-7 ; DD 1-7	No lookahead		
SS ∅ ; DD 1-7	No lookahead		
SS ∅ ; DD ∅	Main lookahead	DMode ∅[H] ∧ $\overline{R7[H]}$	
SS 1-7 ; DD ∅	Secondary lookahead	DMode ∅[H] ∧ $\overline{R7[H]}$ ∧ $\overline{EX3}$	

program during end of same phase

Define a separate Destination Microcode for Double ops

Conditional Branch lookahead = DMode ∅[H] ∧ $\overline{R7[H]}$

Lookahead Logic Added to Control



* - Program flow Data

(During Instruction Start)

(During Instruction Start)

after Instruction Start (ie after EX1 is entered)

(during Same Execution)

[9 words] Double Operand Micro-Code Mode ϕ/ϕ (SS, DD)

Assumes Next instruction Address in AD
Source Register RS, Destination Register RD

MOV B	RS + 0 → DO; DI + SXT → RD; SCC ; BR [BEGIN] V Lookahead
MOV	RS + 0 → RD ; SCC ; BR [BEGIN] V Lookahead
CMP (B)	BE ; RS − RD ; SCC ; BR [BEGIN] + 1
BIT (B)	BE ; RS ∧ RD ; SCC ; BR [BEGIN] + 1
BIC (B)	BE ; $\overline{RS}$ ∧ RD → RD ; SCC ; BR [BEGIN] V Lookahead
BIS (B)	BE ; RS ∨ RD → RD ; SCC ; BR [BEGIN] V Lookahead
ADD	RS + RD → RD ; SCC ; BR [BEGIN] V Lookahead
SUB	RD − RS → RD ; SCC ; BR [BEGIN] V Lookahead

Changes 1 June 1976 — lookahead = D Mode ϕ ∧ RD [L]

time = .2 μsec (Microcycle)

4 March 76
ARB

[9 words] Double Operand Microcode Modes 1-7 / 0
SS DD

Assumes Next instruction Address is in AD
And Source is in T0

MOVB DI+SXT → RD ; SCC ; BR [BEGIN]V(lookahead) ; Secondary lookahead DATIIR

MOV T0+0 → RD ; SCC ; BR[BEGIN]V(lookahead)

CMP(B) BE ; T0-RD ; SCC ; BR [BEGIN]+1 ; DATIIR

BIT (B) BE ; T0∧RD ; SCC ; BR [BEGIN]+1 ; DATIIR

BIC (B) BE ; $\overline{T0}$∧RD → RD ; SCC ; BR [BEGIN]V(lookahead) ;

BIS (B) BE ; T0∨RD → RD ; SCC ; BR [BEGIN]V(lookahead)

ADD T0+RD → RD ; SCC ; BR [BEGIN]V(lookahead) ;

SUB RD-T0 → RD ; SCC ; BR [BEGIN]V(lookahead) ;

PC mode 0 addressing requires BR BEGIN
and Conditional (Lookahead) DATIIR

tcmic = .2 μsec (Microcycle)

requires special lookahead logic mode SS

[8words] Double Operand Microde Modes 0/1-7
SS DD

Assumes Destination Mode executed with
Destination address in AD and Data coming to DI

MOV B NOA WI/O[L]; RS → DO; SBCC; BR[BEGIN]; DATOB

MOV NOA WI/O[L]; RS → DO; SCC; BR[BEGIN]; DATO

CMP (B) special DD WI/O[H]; BE; RS-DI; SCC; ~~R7→AD; BR[BEGIN]; DATI IR~~

BIT (B) special DD WI/O[H]; BE; DI ∧ RS; SCC; ~~R7→AD; BR[BEGIN]; DATI IR~~

BIC (B) WI/O[H]; BE; $\overline{RS} \wedge DI \rightarrow DO$; SCC; BR[BEGIN]; DATO(B)

BIS (B) WI/O[H]; BE; RS ∨ DI → DO; SCC; BR[BEGIN]; DATO(B)

ADD WI/O[H]; RS + DI → DO; SCC; BR[BEGIN]; DATO

SUB WI/O[H]; DI - RS → DO; SCC; BR[BEGIN]; DATO

except MOV(B) { times .2 μsec (microcycle) NO DATO(B)
.5 μsec with DATO(B)

MOV(B) .3 μsec

4 March 1976
ARD

[8 words] Double Operand Microcode Modes 1-7/1-7

Assumes Source executed with Source in TØ [3w
Assumes Destination executed with
Destination Address in AD and Data Comming to DI

MOV B W I/O [L]; TØ → DO; SCC; BR[BEGIN]; DATO B

MOV W I/O [L]; TØ → DO; SCC; BR[BEGIN]; DATO

[5w
CMP (B) W I/O [H]; BE; TØ - DI; SCC; ~~R7 → AD; BR[BEGIN+]; DATI IR~~
Special DD

BIT (B) W I/O [H]; BE; TØ ∧ DI; SCC; ~~R7 → AD; BR[BEGIN+]; DATI IR~~
Special DD

BIC (B) W I/O [H]; BE; $\overline{TØ}$ ∧ DI → DO; SCC; BR[BEGIN]; DATO (B)

BIS (B) W I/O [H]; BE; TØ ∨ DI → DO; SCC; BR[BEGIN]; DATO (B)

[3 w
ADD W I/O [H]; TØ + DI → DO; SCC; BR[BEGIN]; DATO

SUB W I/O [H]; DI - TØ → DO; SCC; BR [BEGIN]; DATO

except MOV (B) times .2 μsec No DATO (B)
.5 μsec with DATO (B)

MOV (B) .3 μsec

OTHER INSTRUCTIONS

[3 words]

RTS R6 → AD; R6+#2 → R6; DATI
W/O L RD → R7 + AD Delayed DATI IR

W I/O [H]; DI → RD; R7 → AD; BR [BEGINX]; DATI IR

time 2 micro + 1 Memory = 1.0 μsec

[5 words]

MARK IR<05:00> → *2 → T0
T0 + R6 → R6 + AD ; DATI
R6 + #2 → R6
W I/O [L] R5 → R7 + AD ; Delayed DATI IR

W I/O [H]; DI → R5; R7 → AD; BR [BEGINX]; DATI IR

time 3 micro + 1 Memory = 1.2 μsec

[3 words]

SOB RS - 1 → RS' SZ'
IR(5:00) * 2 → T0 ; if Z' = 1 BR [BEGIN]
R7 - T0 → R7 → AD; BR [BEGINX] ; DATI IR

time No Branch .4 μsec
w. Branch .6 μsec

5 March 1976
APP

All Branches

[2 words]

include Authentic operation of I-1 statement in DK (ICode) sequence

B___ IR<07:00>+SXT * 2 → TØ ; if Condition Not satisfied BR [BEGIN]
R7 + TØ → R7+AD ; BR [BEGINX] ; DATI IR

times No Branch .2
with Branch .4

[2 words]

CLX CCC ; BR [BEGINX] ; DATI IR

SEX SCC ; BR [BEGINX] ; DATI IR

time .2 μsec all modes

[2 words]

WAIT Repeat Microcycle until [INT] ; Delayed DATIIR (inhibited)
BR [ICODE]

Not used

SPL Load Priority from IR <03:00> ; BR [BEGINX]
dependent on Kernel/User (in user PS not loaded)

Preliminary Console Routine

[2 words]

Kernel/sr RESET — ON USER BR [BEGIN]
[INIT] ; BR [BEGINX] ; DATI IR

[35 words]

User → HALT — # TRAPV → TØ+AD ; BR [ENTER] ; DATI(H)

Kernel → [HALT] (Console Halt) — SWC → TØ ; RØ → DO
Ø → T2
Ø → T3

[GO] — SWC ∧ TØ → T1
SWC → TØ ; BR on Z'=1 [GO]
Rotate Right T1
RR T1 ; on Z'=Ø BR [DEP]
RR T1 ; on Z'=Ø BR [STRT]
RR T1 ; on Z'=Ø BR [LAD]
RR T1 ; on Z'=Ø BR [EX]
BR [BEGINX+1] ; DATI IR

[LAD] — SW1 → AD + T6
SW 2 → ADX + T7 BR [HALT+1]
~~ADØ → AD~~
~~AD1 → ADX ; BR [HALT+1]~~

[STRT] — T7 → PC ; BR [BEGIN] ; INIT

[EX] — T2 → T2
1 → T2 ; on Z'=Ø BR [D2]

[DI] — T6+1 +Reg[Ø] → T6
~~ADØ +/Ø (Regmode) → ADØ~~
T6 + Carry → T7

[D2] — T6 → AD
T7 → ADX
T2 → T2
SWØ → DO ; on Z'=Ø BR [DP]
Ø → T3 ; on Reg Mode BR [OUT1]
BR [GO] ; DATI

[OUT1] — RD → DO ; BR [GO]

[DEP] — T3 → T3
1 → T3 ; on Z'=Ø BR [D2]
BR [D1]

[DP] — Ø → T2 ; on Reg Mode BR [OUT2]
BR [GO] ; DATO

[OUT2] — BR [GO] ; DO → RD

Shaw 1972
APD

TRAP & INTERRUPT ROUTINE

[9 words common] + [1 word for each Vector] (or each Trap address)

↓ seperate for each type of interrupt

[TRAP] or [INT]
[ENTER]

#TRAPV → T0 + AD ; BR [ENTER] ; DATI (H)
T0 + #2 → T0
PC → T2 (or PC - #2 → T2)

New PC: WI/O[H] ; DI → PC ; T0 → AD ; DATI (H)
PS → T1

WI/O[L] ; R6 - #2 → R6 + AD + EXR6
old PS → T1 → DO ; DI → T0 ; DATO (in new stack)
New PS → T0 → PS ;
WI/O[L] ; R6 - #2 → R6 + AD + EXR6
old PC → stack: T2 → DO ; BR [BEGIN] ; DATO

trap & interrupt vectors are always loaded from Kernel space

old PS & PC linkage is 'pushed' onto stack specified by new PS

times [Trap] or [INT]. 2.8 μsec

.2 + .6 + .2 + .6 + .6 + .2 + .3

Must be Multiprocessor include stack check!

Add the Capability of 16 Bit Data Word in the Microcode for Masking & loading interrupt Vectors - use the same bits as the shift/rotate

Changes are as follows (for shorter time / less microcode)

Destination Modes

Mode 3 RX → AD ; RX + #2 → RX ; DATI
[OUT1] WI/O[H] ; DI → AD ; BR [IRCODE] ; FETCH (SL)

Mode 5 RX - #2 → RX + AD ; BR [OUTI] ; DATI

Mode 6 R7 → AD ; R7 + #2 → R7 ; DATI
WI/O[H] ; DI + RX → AD ; BR [END] ; FETCH (SL)

mode 7 R7 → AD ; R7 + #2 → R7 ; DATI
WI/O[H] ; DI + RX → AD ; BR [OUT1] ; DATI

Saves 4 words ; Mode 5 time less .2 μsec

Source Modes

Mode 3 RX → AD ; RX + #2 → RX ; DATI
[OUT1] WI/O[H] ; DI → AD ; BR [END1] ; FETCH

mode 5 RX - #2 → RX + AD ; BR [OUT1] ; DATI

mode 6 R7 → AD ; R7 + #2 → R7 ; DATI
WI/O[H] ; DI + RX → AD ; BR [END] ; FETCH

mode 7 R7 → AD ; R7 + #2 → R7 ; DATI
WI/O [H] ; DI + RX → AD ; BR [OUT 1] ; DATI

Saves 4 words ; mode 5 time less .2 μsec

5 March 1976
AFD

[4 words] Other Instruction Microcode Changes

JSR (special)

```
WI/O[L]; R6 - #2 → R6 + EXR6 + AD
DI → TØ ; RX → DO                    ; DATO
R7 → RX + AD ; Delayed DATI IR
TØ → R7 ; BR [BEGIN]X
```

WI/L

Saves 2 words less .4 μsec

Modifs to include Stack check

[11 words]

TRAPS or Instruction TRAPS

~~[RESERVED] # 0 → T0 + AD ; BR [STOP] ; DATI(H)~~

[ILL] or [TIMEOUT] # 4 → T0 + AD ; BR [ENTER] ; DATI(H)

[RESERVED INST] # 10 → T0 + AD ; BR [ENTER] ; DATI(H)

BPT # 14 → T0 + AD ; BR [ENTER] ; DATI(H)

IOT # 20 → T0 + AD ; BR [ENTER] ; DATI(H)

[POWER FAIL] # 24 → T0 + AD ; BR [ENTER] ; DATI(H)

EMT # 30 → T0 + AD ; BR [ENTER] ; DATI(H)

TRAP # 34 → T0 + AD ; BR [ENTER] ; DATI(H)

[MEMORY MGMT] # 250 → T0 + AD ; BR [ENTER] ; DATI(H)

[INTERRUPTS] WI/O[H] D0 → T0 + AD ; BR [ENTER] ; DATI(H)

[BUS ERROR INTRAP] BR [HALT (KERNEL)]

[RESERVED] # 4 → R6 ; BR [TIMEOUT]
Fatal

6 March 1975
ARO

8words INTERRUPT RETURNS

[RTI] R6 → AD; R6 + #2 → R6; DATI
WI/O [L]; R6 → AD; R6 + #2 → R6
DI → R7; DATI
WI/O [H]; DI → R6; R7 → AD; BR [BEGINx]; DATI IR
PSR

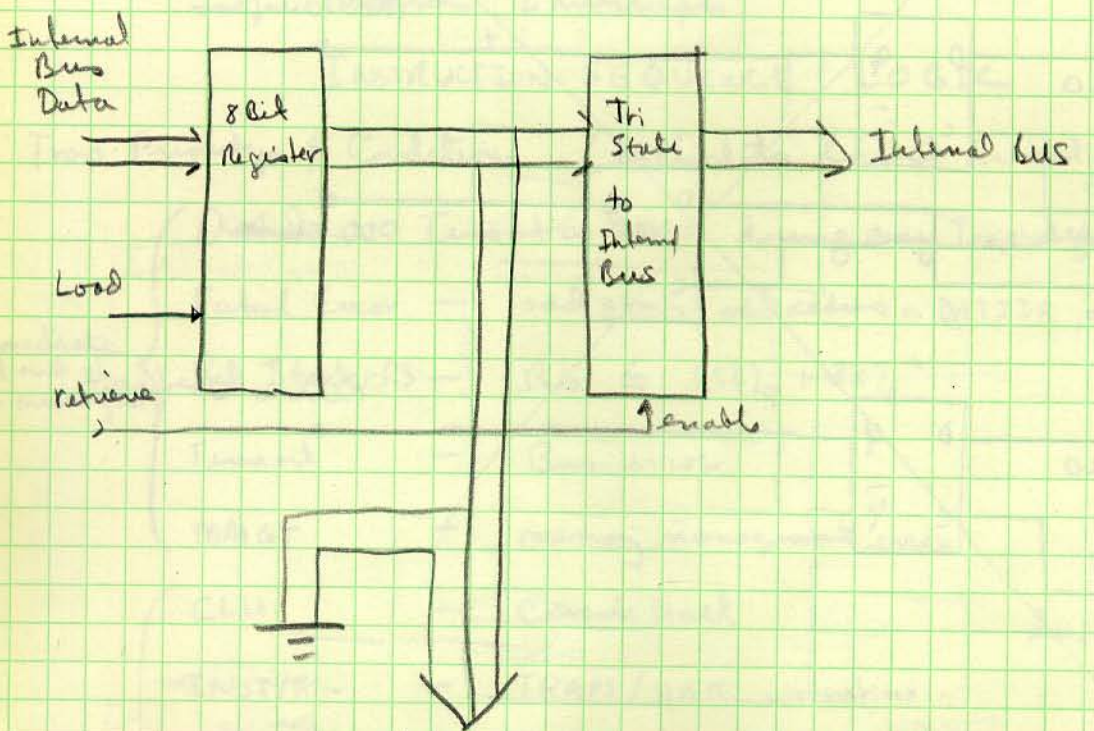
[RTT] R6 → AD; R6 + #2 → R6; DATI
WI/O [L]; R6 → AD; R6 + #2 → R6
DI → R7; DATI
WI/O [H]; DI → R6; R7 → AD; BR [BEGINx]; INHIBIT; DATI IR
PSR

Current Total Words Allocated

Same Mode	14
Destination	13
Single OPs	17 + 16
Dual OPs	18 + 16
Begin	3
Others	19
HALT/Console	35
TRAPS + Vectors	9 + 11
JSR	4
RTI/RTT	8
	140 + 43 = 183 Words

```
                    +R6
        R6 + #2 → AD ; DATI
W1/0    D → PSR
        R6 - #2 — AD ; DATI ; JSR on T=
```

Stack limit Option
(General)



Micro Code Checks stack after Modes 1, 2, 4, & 6
JSR's and TRAPS

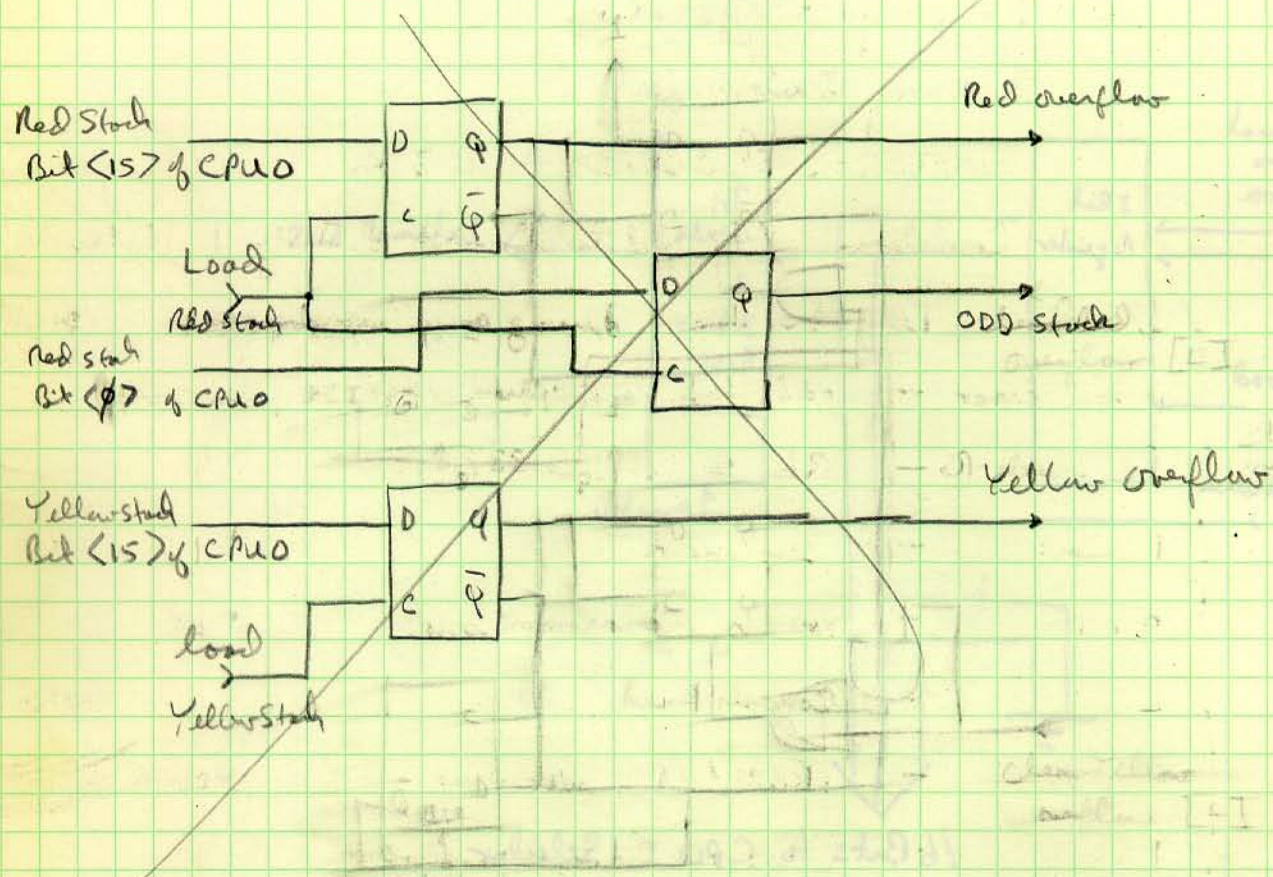
R6 - STACHLIMIT → T1
T1 - #340 → T1 + Red Stack Check
T1 - #40 → Yelow Stack Check

Red and Yelow stack checks are 1 bit Registers which indicate overflow only after one occurs and cannot recur unless stack overflow is first removed.

10 March 1976
ARD

Preliminary

Stack Yellow & Red Check Registers



If Red overflow occurs Yellow overflow is cleared

Red overflow if stack $\leq (SL)_8 + (337)_8$

Yellow overflow if stack $\leq (SL)_8 + (377)_8$

bit <0> of Red stack indicates ODD stack!

if stack errors are cleared by other Trap handlers

stack overflows are cleared!

Preliminary Design of

Force Branch Traps
+
Program Interrupts
+
INSTRUCTION SEQUENCE LOGIC

Trap Priority & Conditions (highest to lowest priority) 16 levels

immediate at end of next microcycle

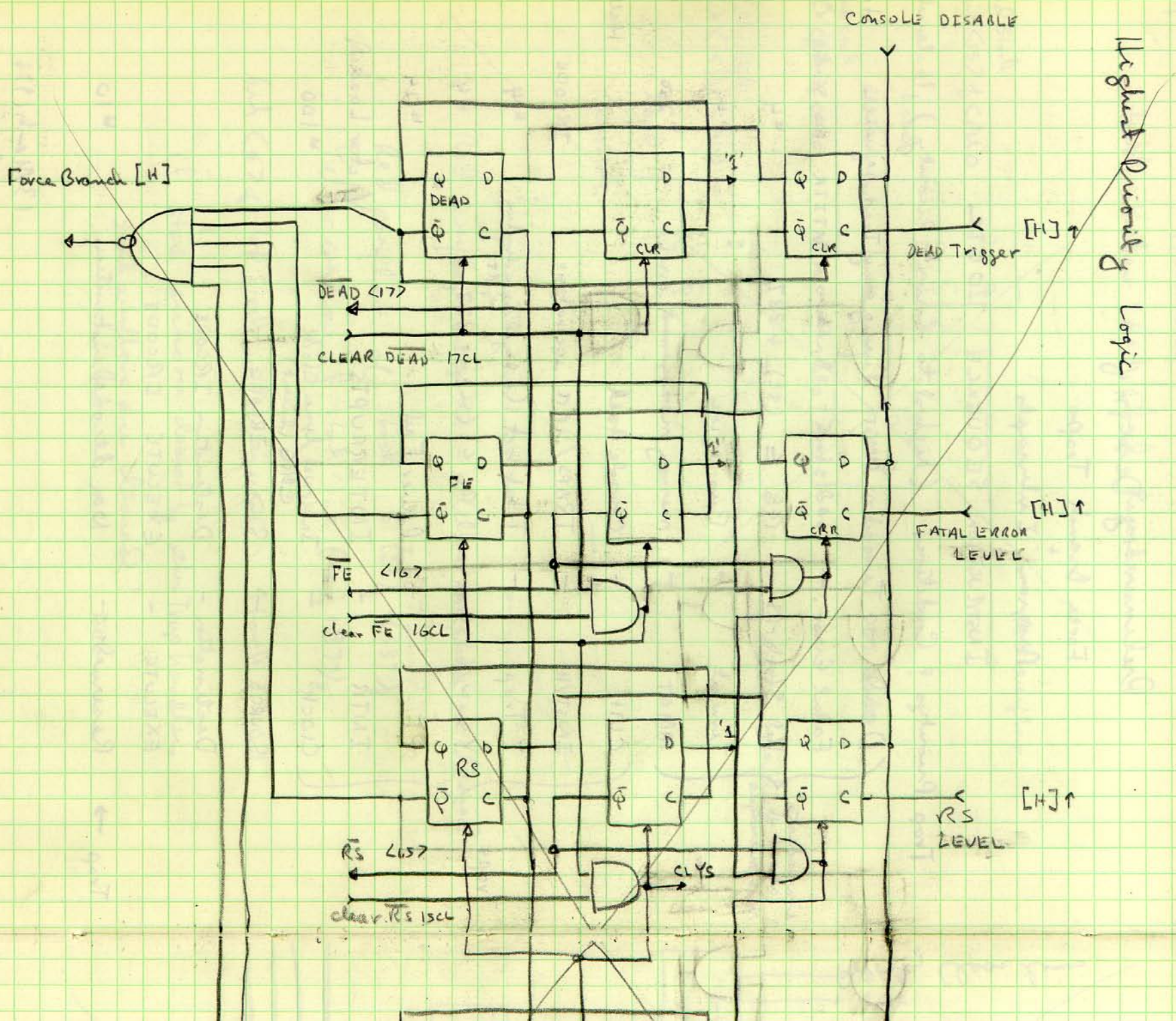
Dead	—	Timeout or MMGT during any Trap sequence		Halt
Fatal Error	—	odd stack, odd address on DATIIR, No stack		#4
Red Stack-RS	—	R6 ≤ $(SL)_8$ + 337_8		#4
Timeout	—	Bus error		#4
MMGT	—	memory management error		#250

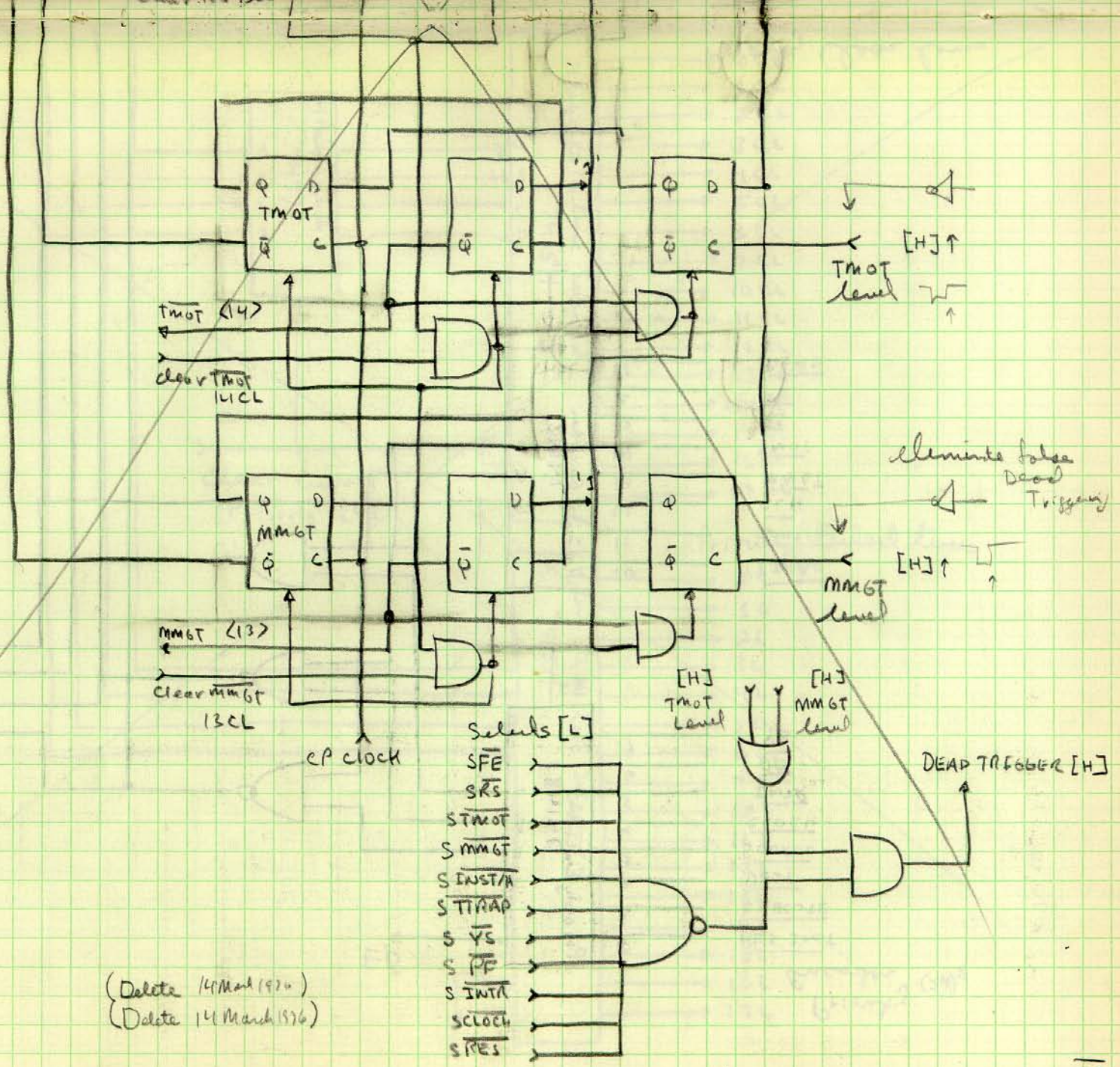
Traps

CLH	—	Console Halt		Halt
INST/H	—	TRAPS / HALT instructions	IRCODE	
T Trap	—	T bit set (delayed one instruction by RTT)		#14
YS-Yellow Stack	—	R6 ≤ $(SL)_8$ + 377_8		#4
PF	—	Power Fail		#24
INTR	—	INTERRUPTS	Vector Loaded	
CLOCK	—	Internal line Clock @BRG (lowest)		#100
SOURCE	—	Source IRCODE		
Destination	—	Destination IRCODE		
EXECUTE	—	EXECUTE IRCODE		

Trap →	ReservedInst	Non decoded instructions	#10

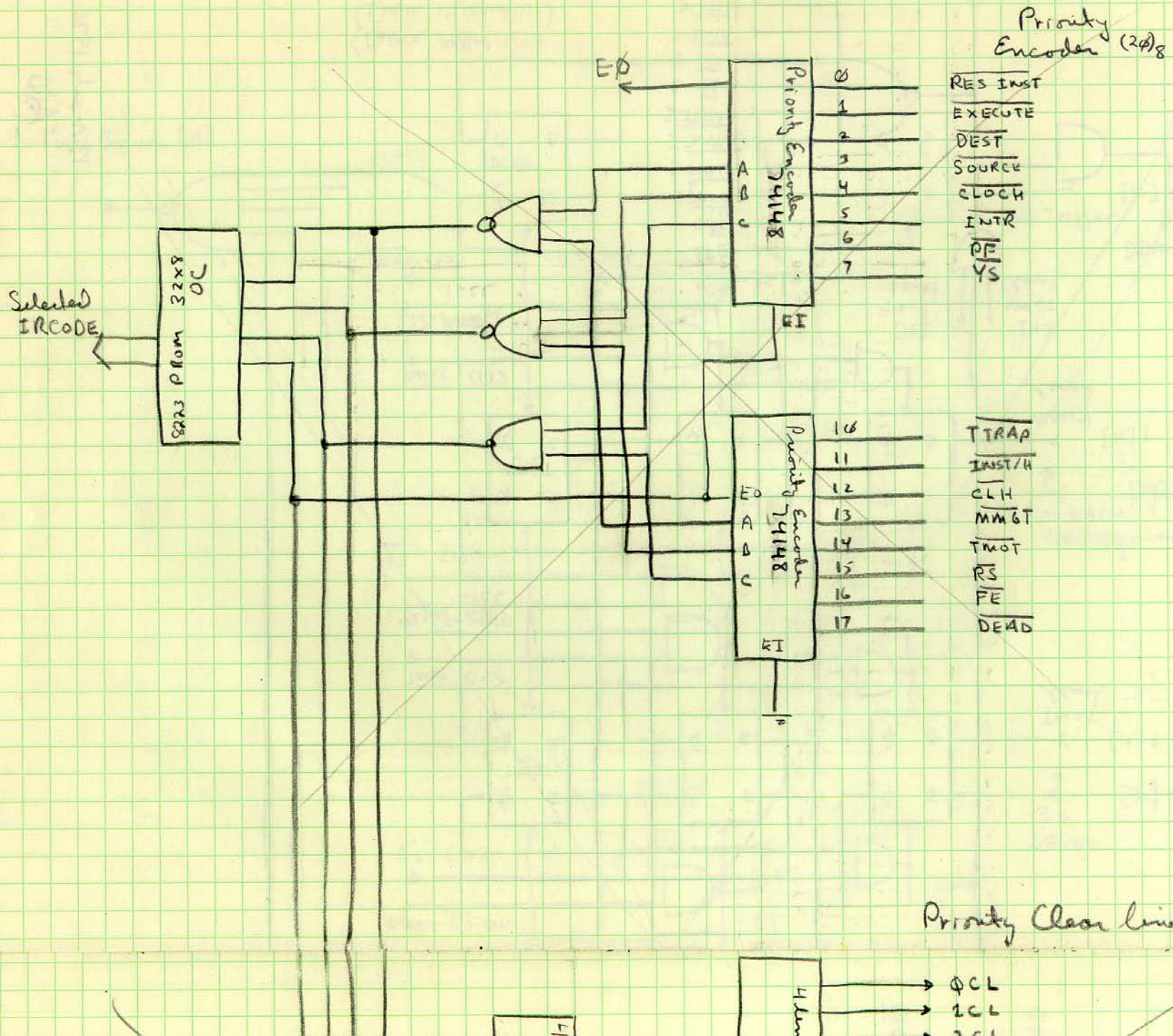
13 March 1976
AMB

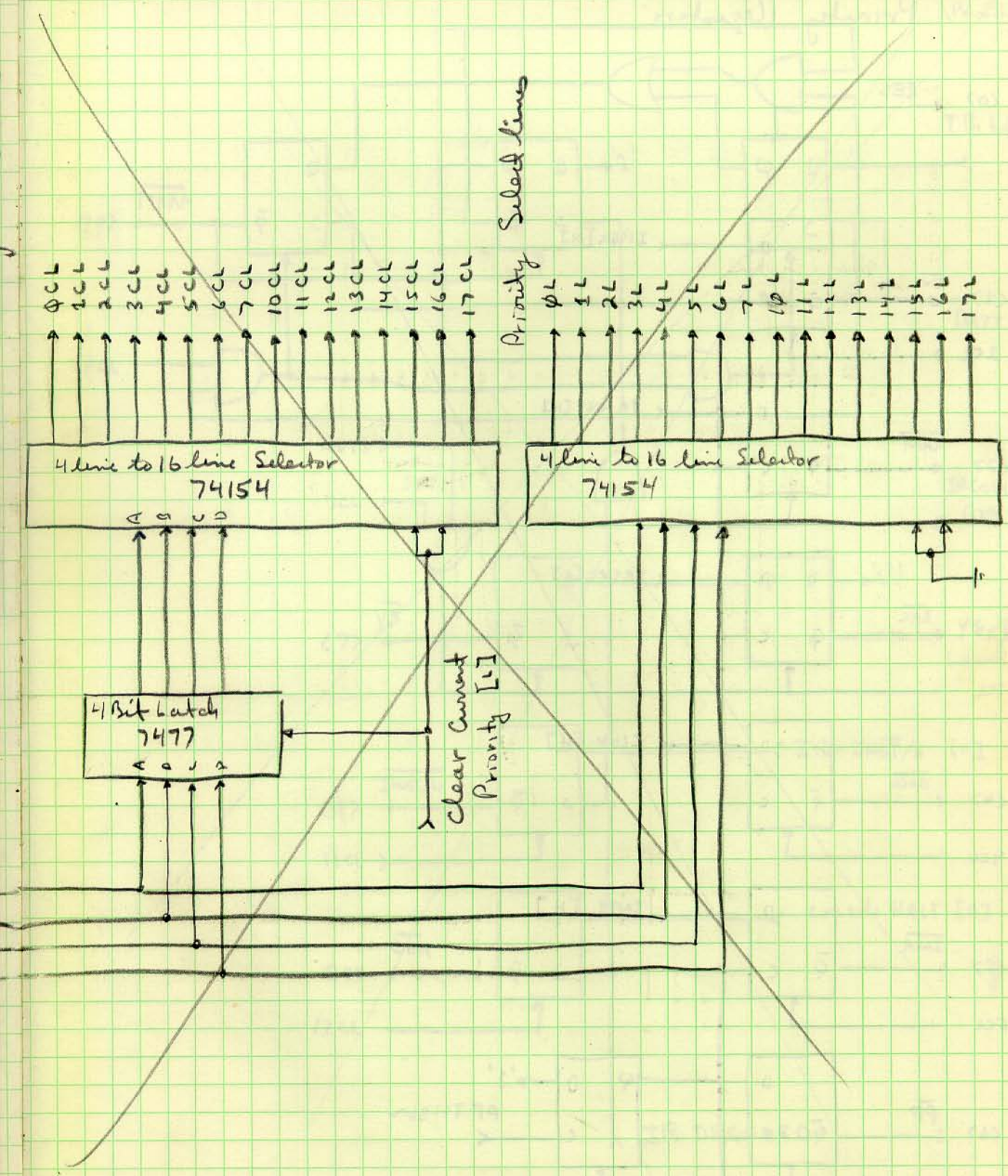

Highest Priority Logic
CONSOLE DISABLE
Force Branch [H]
Q D
DEAD
C
'1'
CLR
DEAD Trigger
[H]↑
DEAD <17>
CLEAR DEAD 17CL
FE
FATAL ERROR LEVEL
FE <16>
clear FE 16CL
Rs
RS LEVEL
RS <15>
CLYS
clear RS 15CL



(Delete 14 March 1976)
(Delete 14 March 1976)

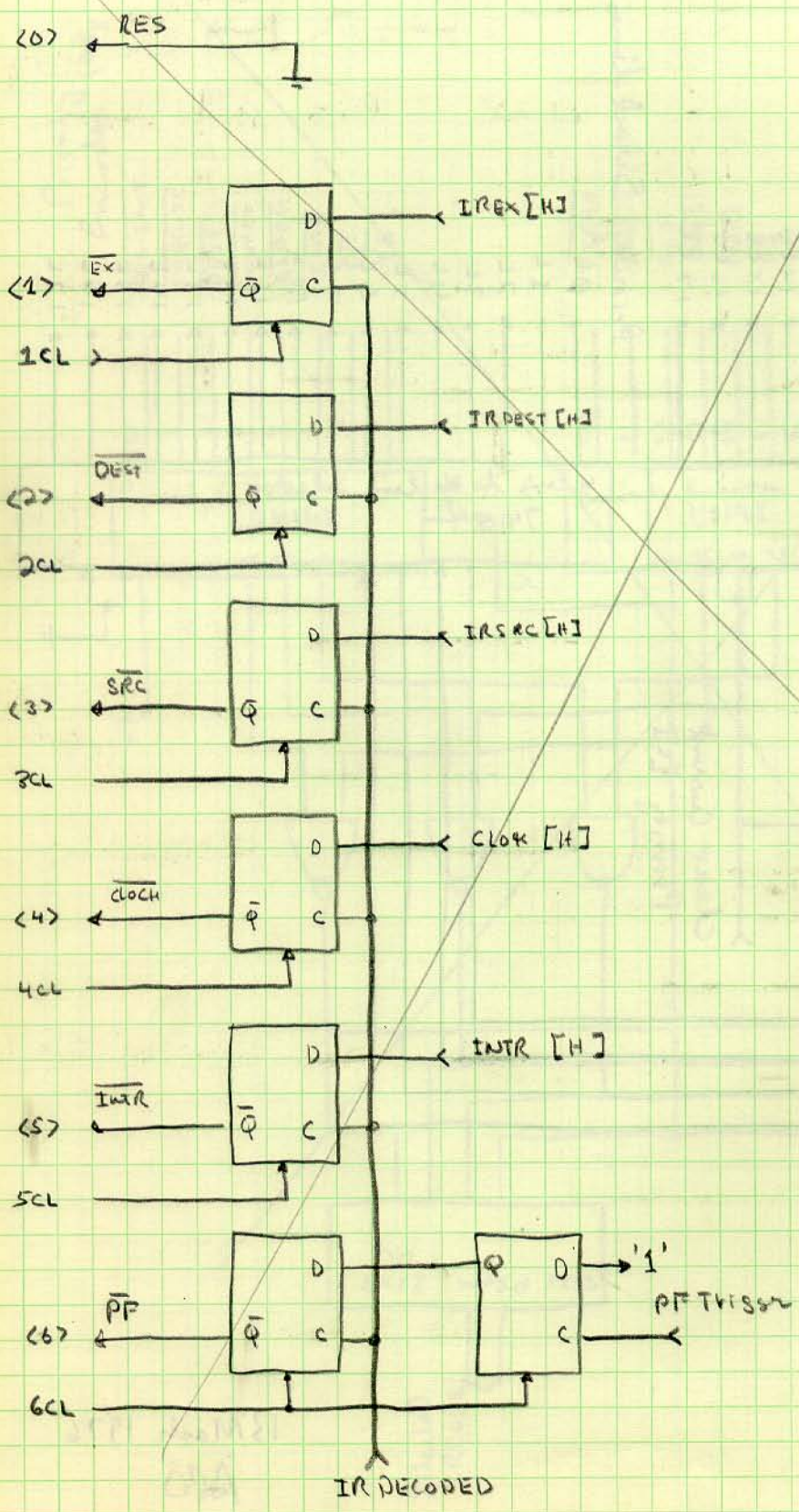
13 March 1976
ARD


Priority Encoder (2Φ)8
EO
Priority Encoder 74148
A
B
C
EI
0
1
2
3
4
5
6
7
RES INST
EXECUTE
DEST
SOURCE
CLOCH
INTR
PF
YS
Selected IRCODE
8223 PROM 32x8 OC
Priority Encoder 74148
EO
A
B
C
EI
16
11
12
13
14
15
16
17
TTRAP
INST/H
CLH
MMGT
TMOT
RS
FE
DEAD
Priority Clear lines
ΦCL
1CL

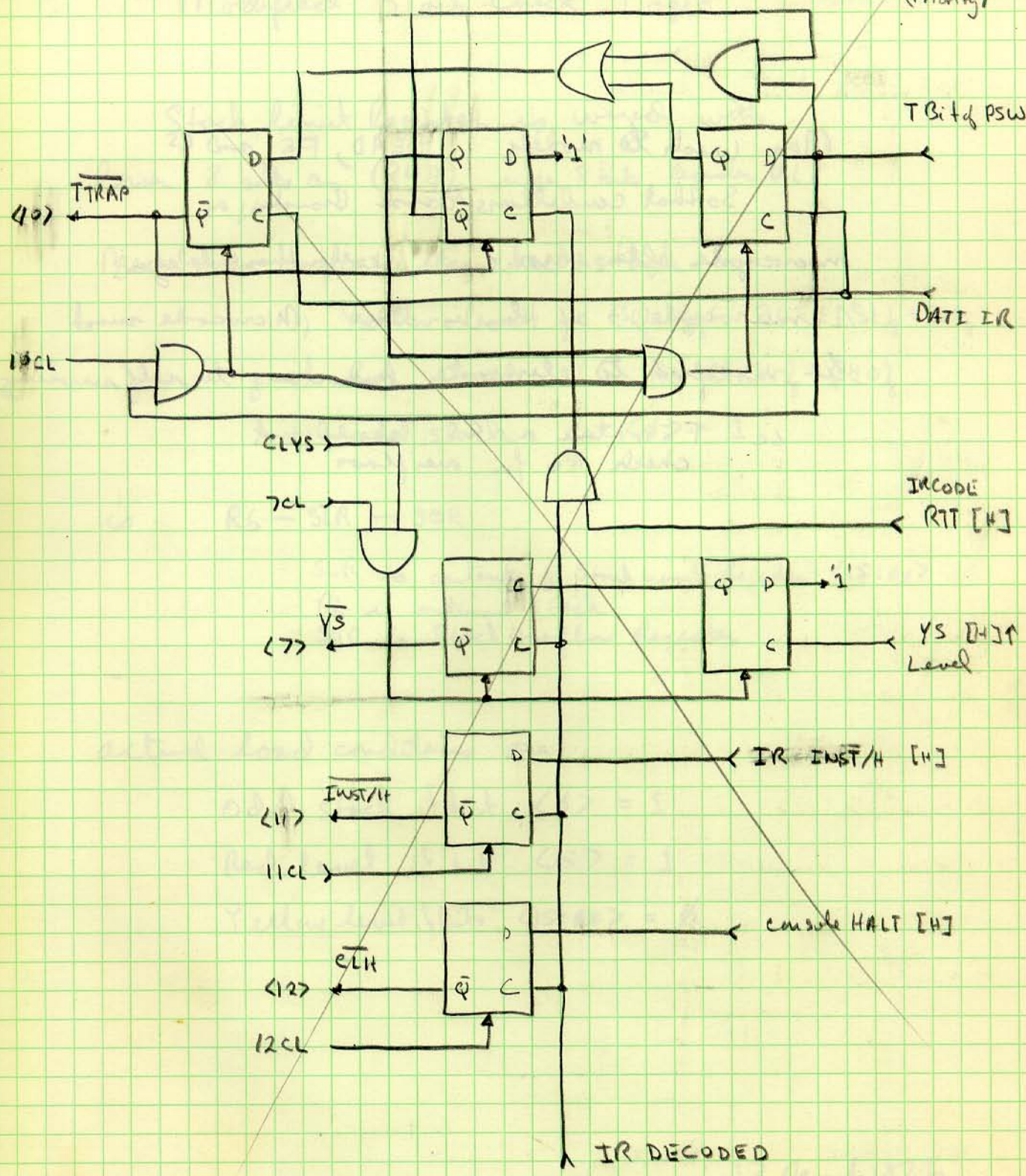


13 March 1976
ARD

Priority Registers



(27 IC's)
(Priority)



13 March 1976
ARD

May wish to modify DEAD, FE, and RS
So that conditions cause Branching on
microcycle after error cycle (rather than delaying
1 microcycle - if this is done Microcode must
be modified to eliminate branching to self servicing
ie FE return or RS should not
check R6 for overflows

Modified STack Check Logic

Stack limit Register is wired with lower 8 bits as $(340)_8$ upper 8 bits Register (SL)

Microcode sequence to check stack for

1) Yellow level i.e $(SL)_8 + 340_8 \leq R6 \leq (SL)_8 + 377_8$
2) Red level $R6 < (SL)_8 + (340)_8$
3) odd stack i.e $\langle 0 \rangle = 1$

is R6 − SLR → SOR

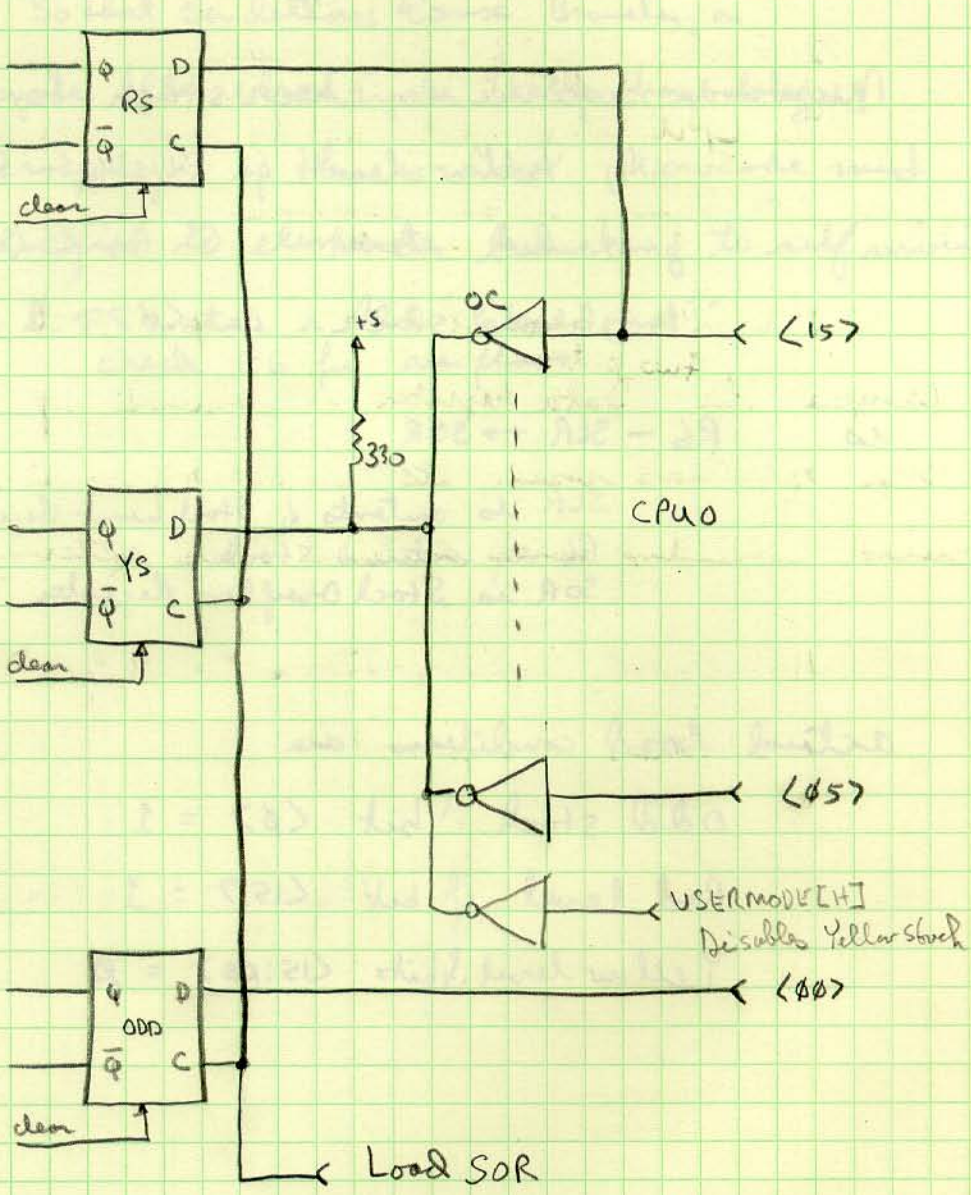
SLR is contents of Stack Limit Register <15:00>
R6 is internal stack
SOR is Stack Overflow Register

actual load conditions are

odd stack if bit $\langle 0 \rangle = 1$
Red level if bit $\langle 15 \rangle = 1$
Yellow level if bits $\langle 15:05 \rangle = 0$

13 March 1976
AFB

Stack check Hardware



CPU External Data Register / IO Data Register D
and Data BUS interface logic

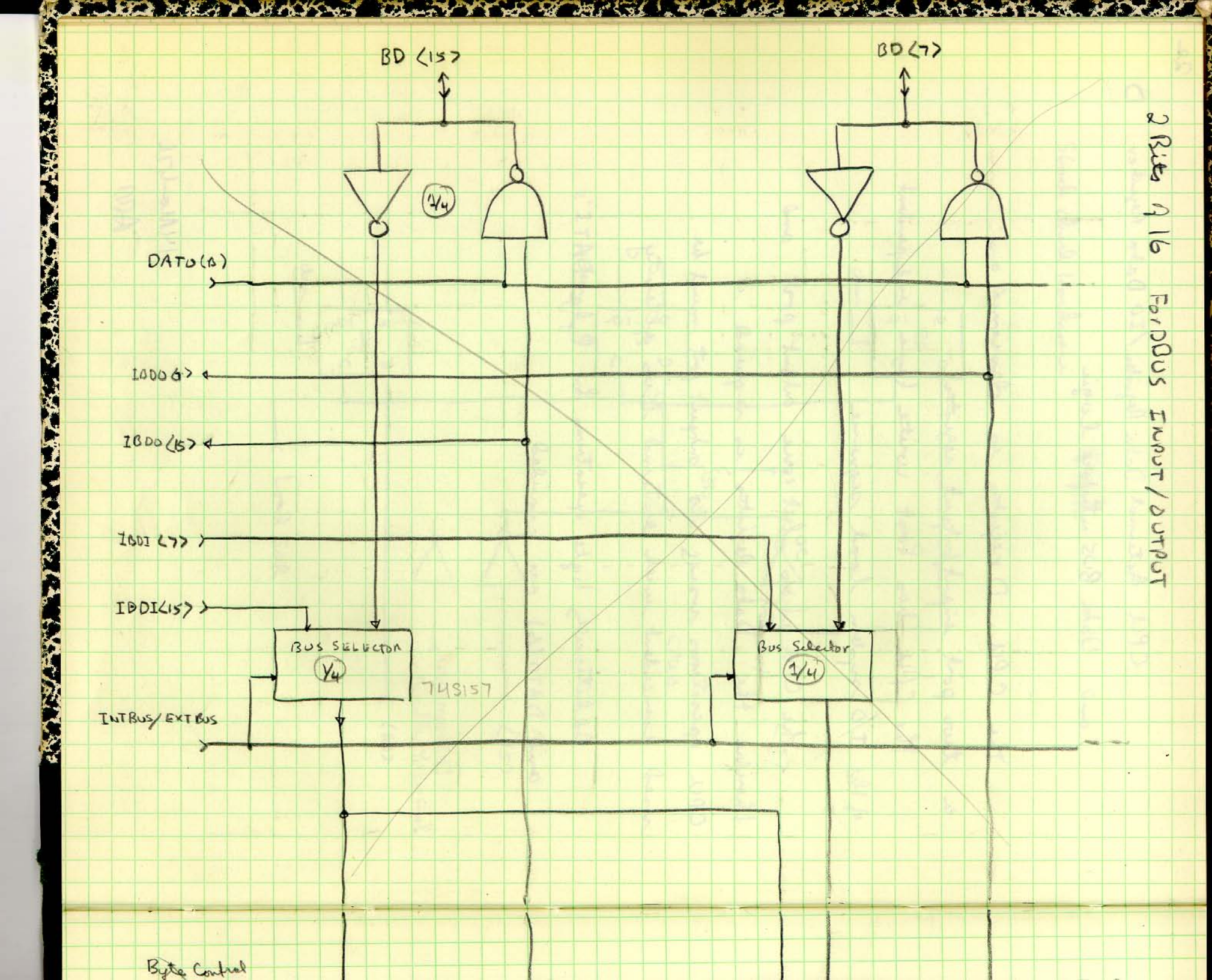
The CPU D register is designed as a two port input/output register -

The CPU has fast write time independent of the I/O system load sequence

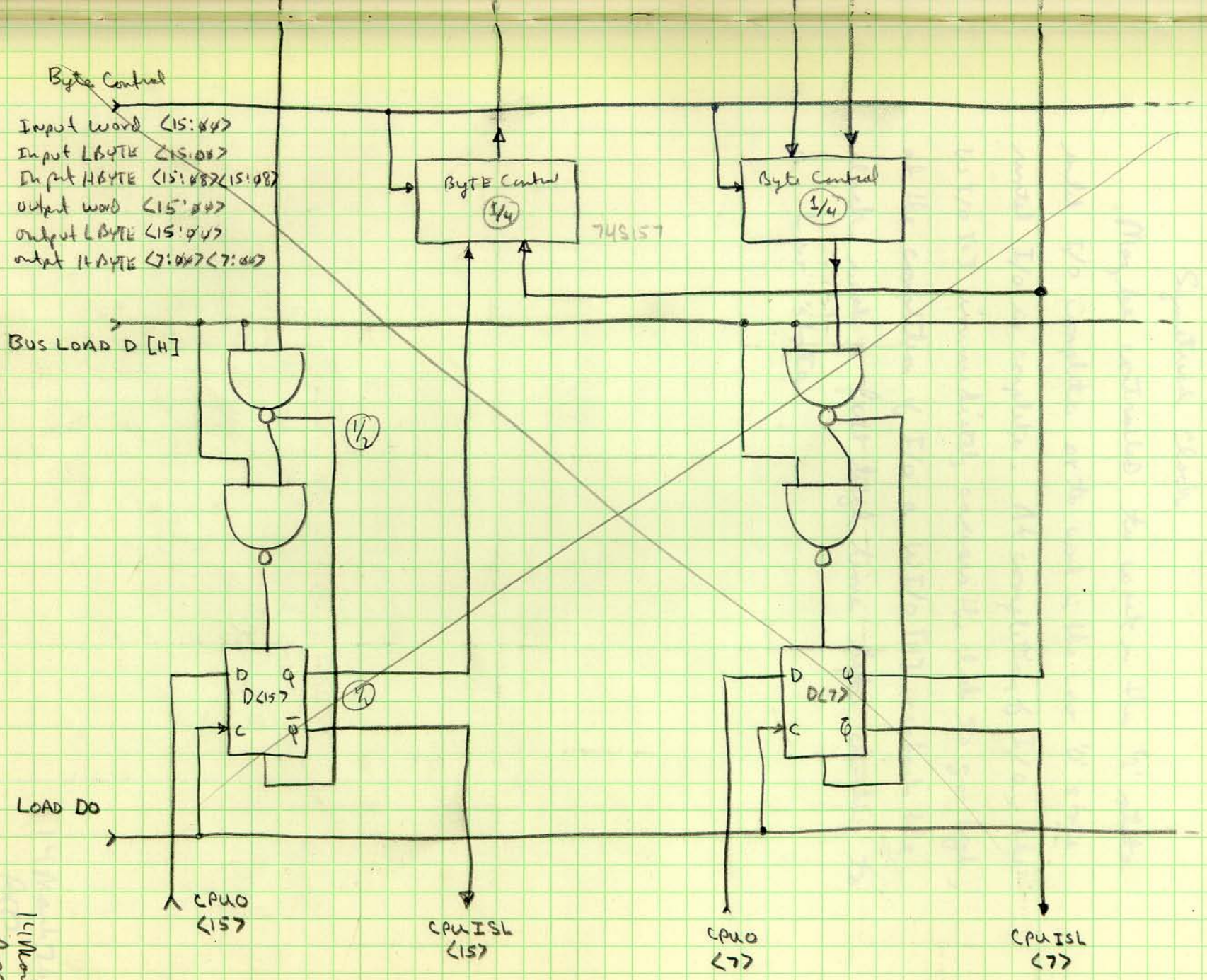
The CPU has but one output port and therefore the two port Data Register is required if CPU operations using its output port must be used coincident with external Bus activity

Additionally byte operations for Byte DATI's and DATO(B) are provided

14 March 76
ARD



2 Bits of 16 For DOUS INPUT/OUTPUT



ADD 16 March 76

Preliminary CPU Clock

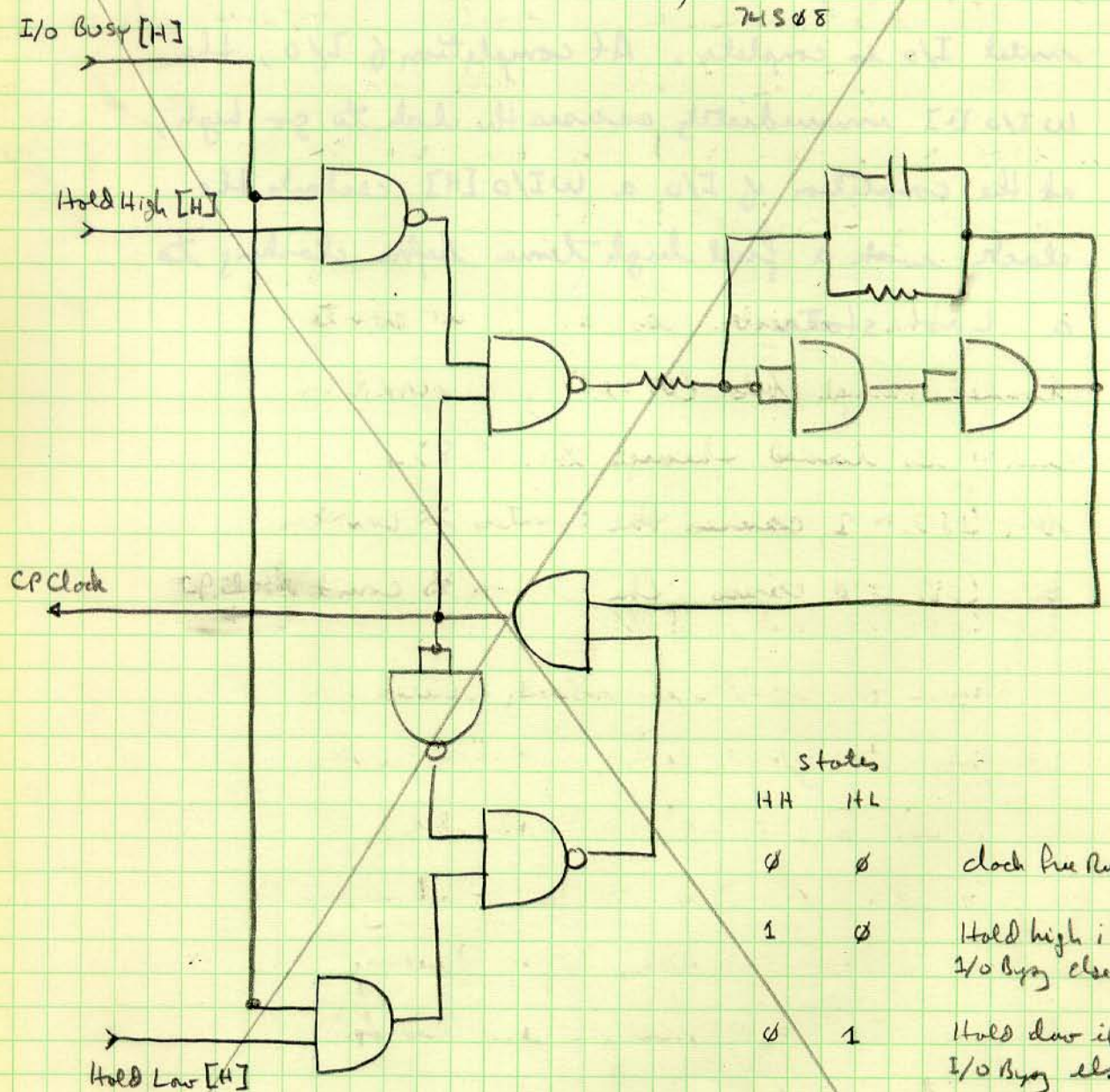
Symetrical Clock

May be controlled to wait in the '1' state until I/O Complete or to wait in the Low '0' state until I/O is complete. At completion of I/O, the WI/O [L] immediately causes the clock to go high, at the completion of I/O a WI/O [H] restarts the clock with a full high time before clocking to a Low state.

14 March 76
AD

CPU clock

Schottky 74S04 and 74S08



States HH	HL	
∅	∅	clock free Run
1	∅	Hold high if I/O Busy else free Run
∅	1	Hold low if I/O Busy else free Run
1	1	free run on I/O not Busy else ?

Step Counter Logic

(Extended Arithmetic)

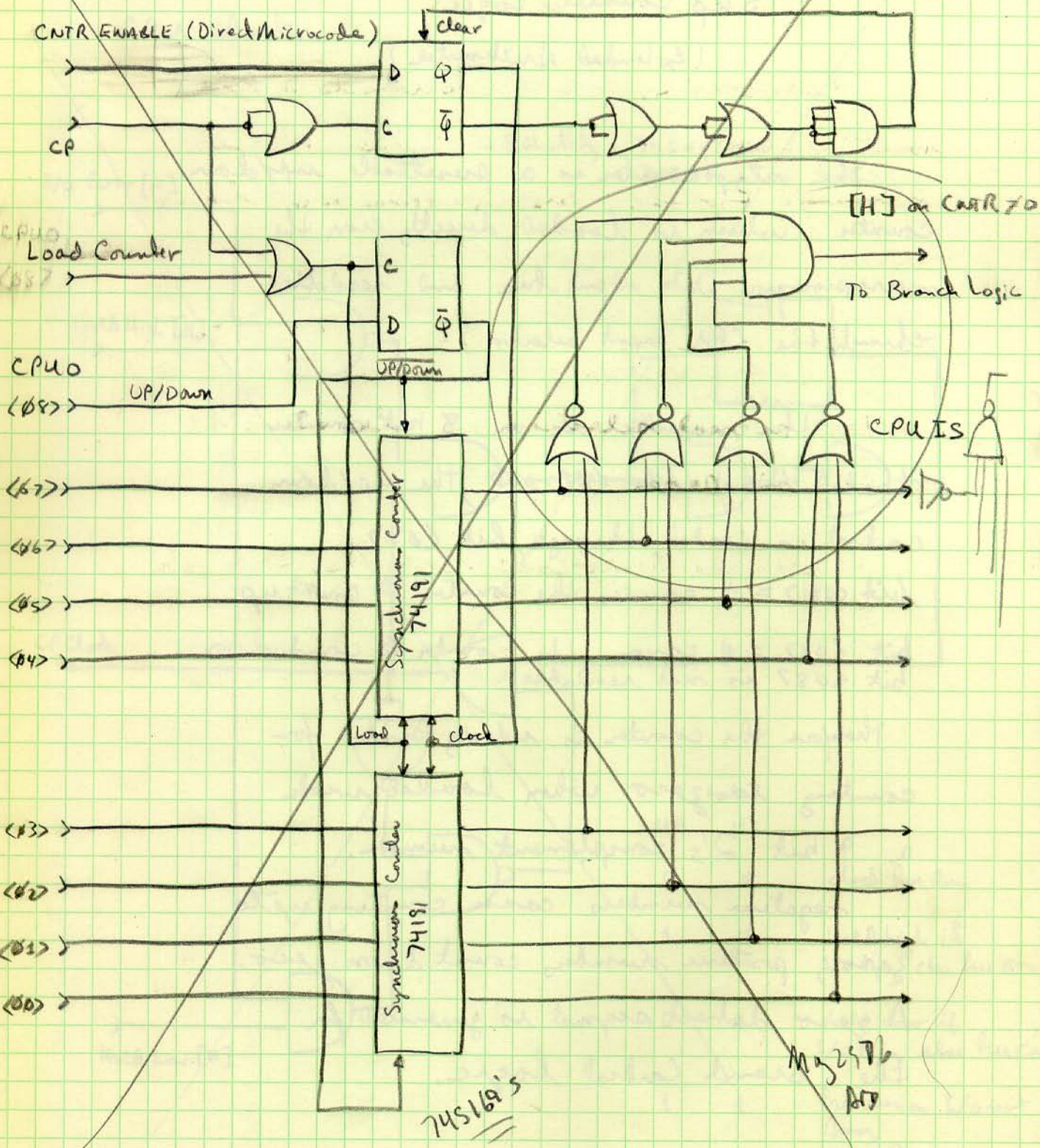
The step counter is a presettable up/down counter which is loaded directly from the microprocessor Data output bus and readable through the CPU Input Selector

the register is an 8 bit counter loaded through bits <07:00>, the up/down control is loaded through bit <08>.
bit <08> = 1 causes the counter to count up
bit <08> = 0 causes the counter to count down
bit <08> is not readable

Therefore the counter is ideally suited for counting to zero when loaded with a 9 bit 2's complement number.

negative numbers cause counting up to zero, positive numbers count down zero.
A zero detect signal is generated for the Branch Control logic.

14 March 76
ARB


CNTR ENABLE (Direct Microcode)
clear
D
Q
C
CP
Load Counter
[H] on CNTR=0
To Branch Logic
CPUIS
CPUO
UP/Down
UP/Down
Synchronous Counter 74191
Load
clock
Synchronous Counter 74191
74S169's
M3276
ALU

CPU IS Logic

is composed of 16 16 line to 1 line multiplexers to provide 16 source operands for the microprocessor inputs lines

the selectors are 74150 24pin Ic's

the signal selections are listed on the following pages.

4 Bits of the Micro-code select the input Data

14 March 1976
ARD

Preliminary Selection code

DCBA				
0000	D register		<15:00>	word
0001	D register	<07:00>	<15:08>	SWAB
0010	D register		<00:15>	Reverse
0011	D register	[07]	<07:00>	SxT Byte
0100	D register	[05]	<05:00>	SxT 6 Bits
0101	I register		<15:00>	word
0110	I register	[07]	<07:00>	SxT Byte
0111	I register	-0-	<05:00>	Low 6 Bits
1000	Scratch Registers		<15:00>	
1001	Priority Vector	-0-	<7:00>	
1010	Processor Status		<15:00>	
1011	Stack Limit Register +$(340)_8$		<15:00>	
1100	STop Counter	-0-	<07:00>	
1101	Microcode Constant		<15:00>	
1110	Console Switch 1		<15:00>	Add <15:04>
1111	Console Switch 2		<15:00>	Cntrl <15:04> Ad <05:00>

* Not necessary for Emulation — Might Replace with

~~SLR$_8$ + $(342)_8$~~
SLR$_8$ + $(344)_8$

✓ VECTOR PROM (must)

✓ add <15:04> → <00:15>

✓ Scratch registers

CPU Micro Controller Logic

The Micro Controller sequences the CPU through the micro program by controlling the branching sequence in the CPU operation

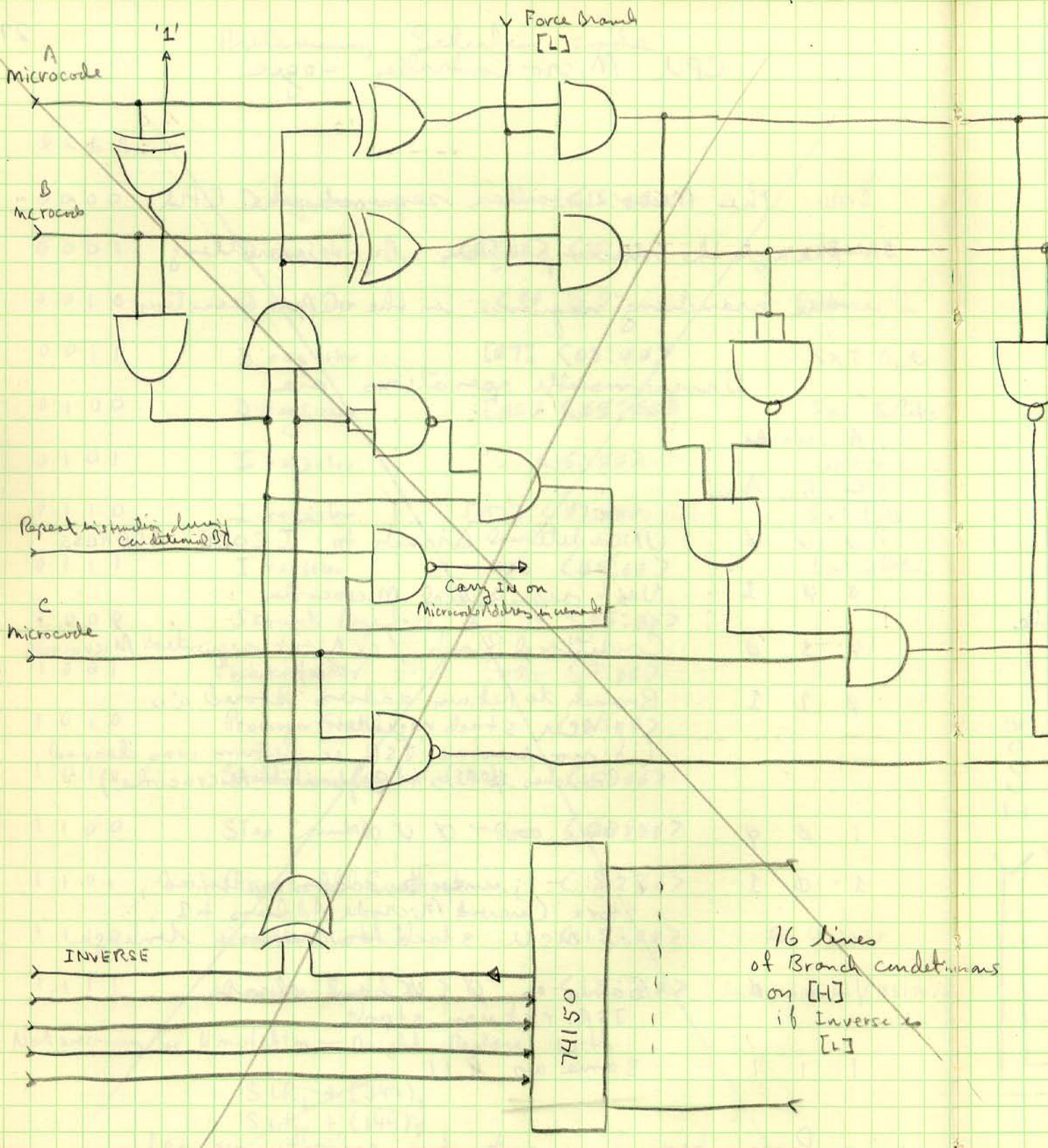
Programmable operations are

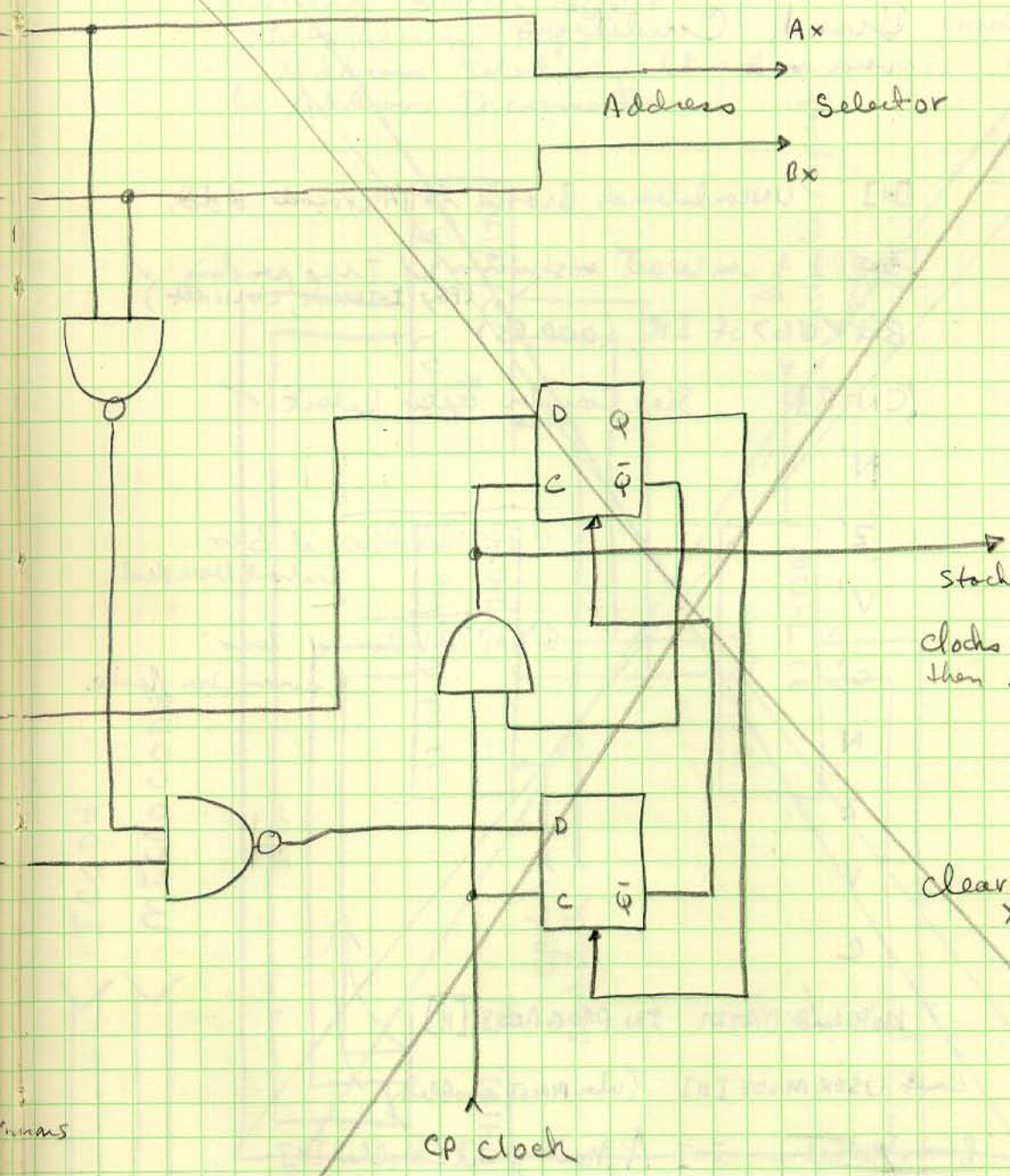
Microcode

C	B	A	
ø	ø	ø	UNconditional Branch to ICODE ADDRESS
ø	ø	1	Next sequential Microcode
ø	1	ø	Conditional Branch / or Next sequential Microcode
ø	1	1	Branch to Return address stored in MCU stack register (if no previous JSR or Return was cleared Branches to Next Sequencial Microcode)
1	ø	ø	Same as ø ø ø
1	ø	1	JSR – jump to address specified, store Current Microde Address +1 on MCU stack as return linkage
1	1	ø	Same as ø 1 ø and clears JSR return stack
1	1	1	Same as ø 11

a force branch input has priority over all modes and forces an unconditional Branch to Icode, a JSR will be aborted / a programmed stack clear will occur

15 Mar 76
ARS


'1'
Force Branch
[L]
A
Microcode
B
Microcode
Repeat instruction during conditional BR
Carry IN on
Microcode Address incrementer
C
Microcode
INVERSE
74150
16 lines
of Branch conditions
on [H]
if Inverse is
[L]



0	0	ICODE
0	1	NEXT
1	0	BR micro
1	1	Return

15 March 76
AFD

Conditional Branch Conditions

on E = 0 (inverse is E = 1)

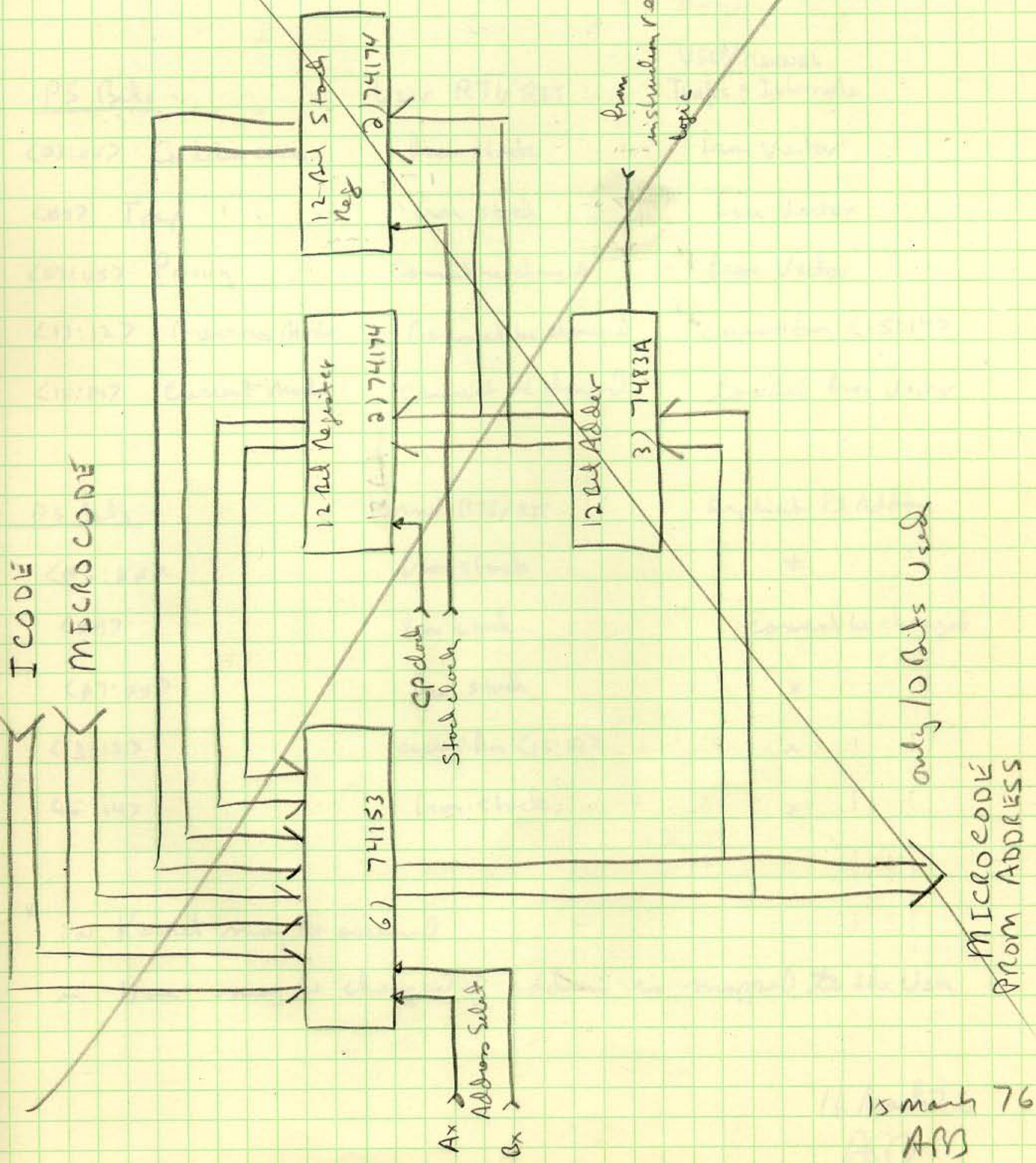
D	C	B	A	
0	0	0	0	[H] Unconditional Branch to Microcode Addr,
0	0	0	1	[INT] in Wait instruction if Trap pending (PF, INTERRUPT, CNZ1HALT) ✓
0	0	1	0	Bit <06> of IR (ODD RS) ✓
0	0	1	1	[CNTR] Step Counter Zero Detect ✓
0	1	0	0	N'
0	1	0	1	Z'
0	1	1	0	V'
0	1	1	1	C'
1	0	0	0	N
1	0	0	1	Z
1	0	1	0	V
1	0	1	1	C
1	1	0	0	LookAhead DATIN IN PROGRESS [H] ✓
1	1	0	1	Curr USERMODE [H] (when MMGT enabled)
1	1	1	0	Previous Mode User [H] ∧ Management Enabled [H]
1	1	1	1	Register Mode [L] ie Addr is 7777[11x][xxx] ✓

$\overline{\text{Ø37776 VPSW}}$ if Curr User Mode

$\overline{\text{147776 VPSW}}$ if Prev User Mode

I CODE

MCU Stack Register
Next Address Register
Address Selector
& Address Incrementer



15 march 76
ABB

Processor Status Register Logic

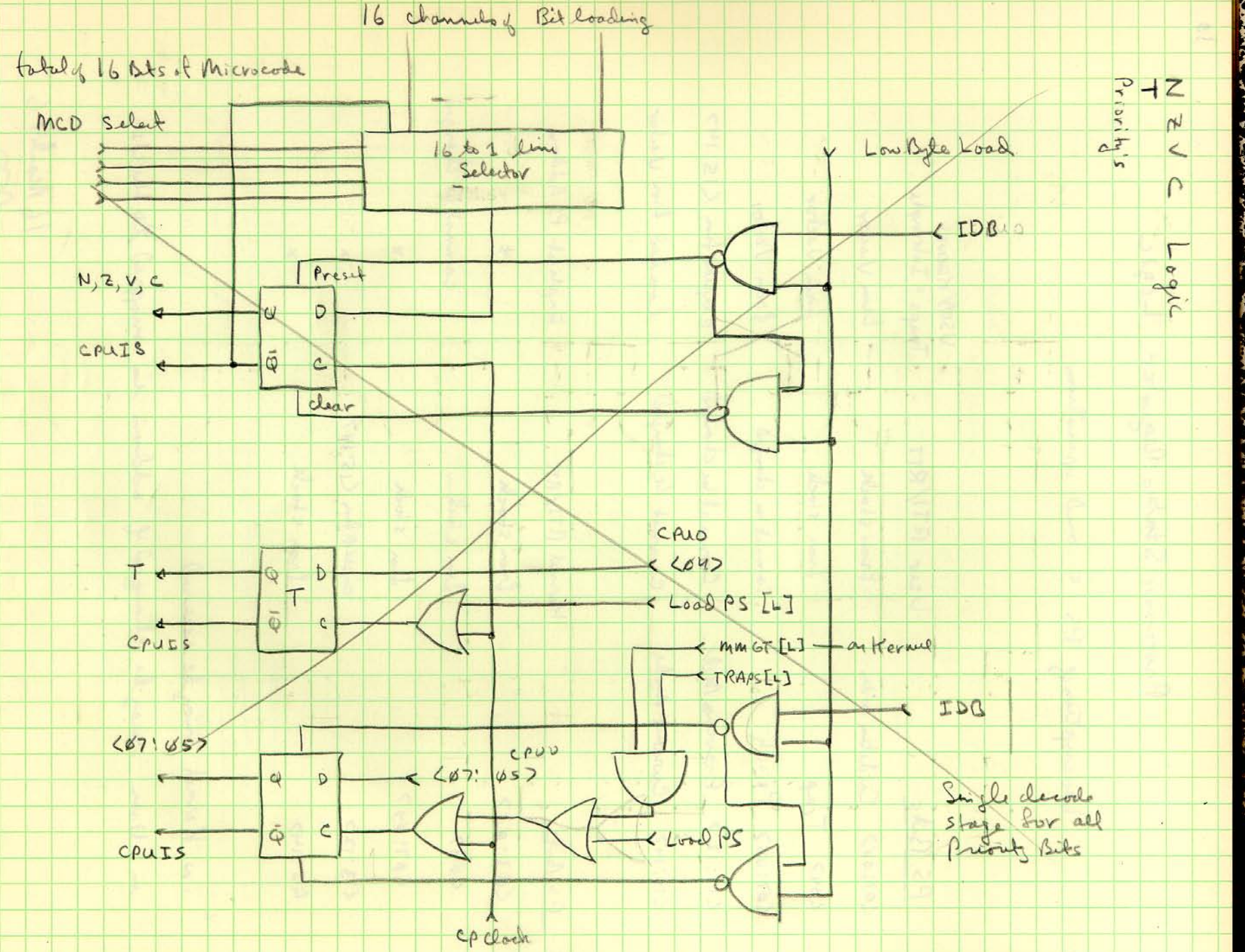
Description of PS & and operations

PS Bits	User RTI/RTT	USER/KERNEL Traps & Interrupts
<03:00> Condition codes	from stack	from Vector
<04> Trap	from stack	from Vector
<07:05> Priority	Cannot be changed	from Vector
<13:12> Previous Mode	Cannot be changed	copied from <15:14>
<15:14> Current Mode	cannot be changed	loaded from Vector

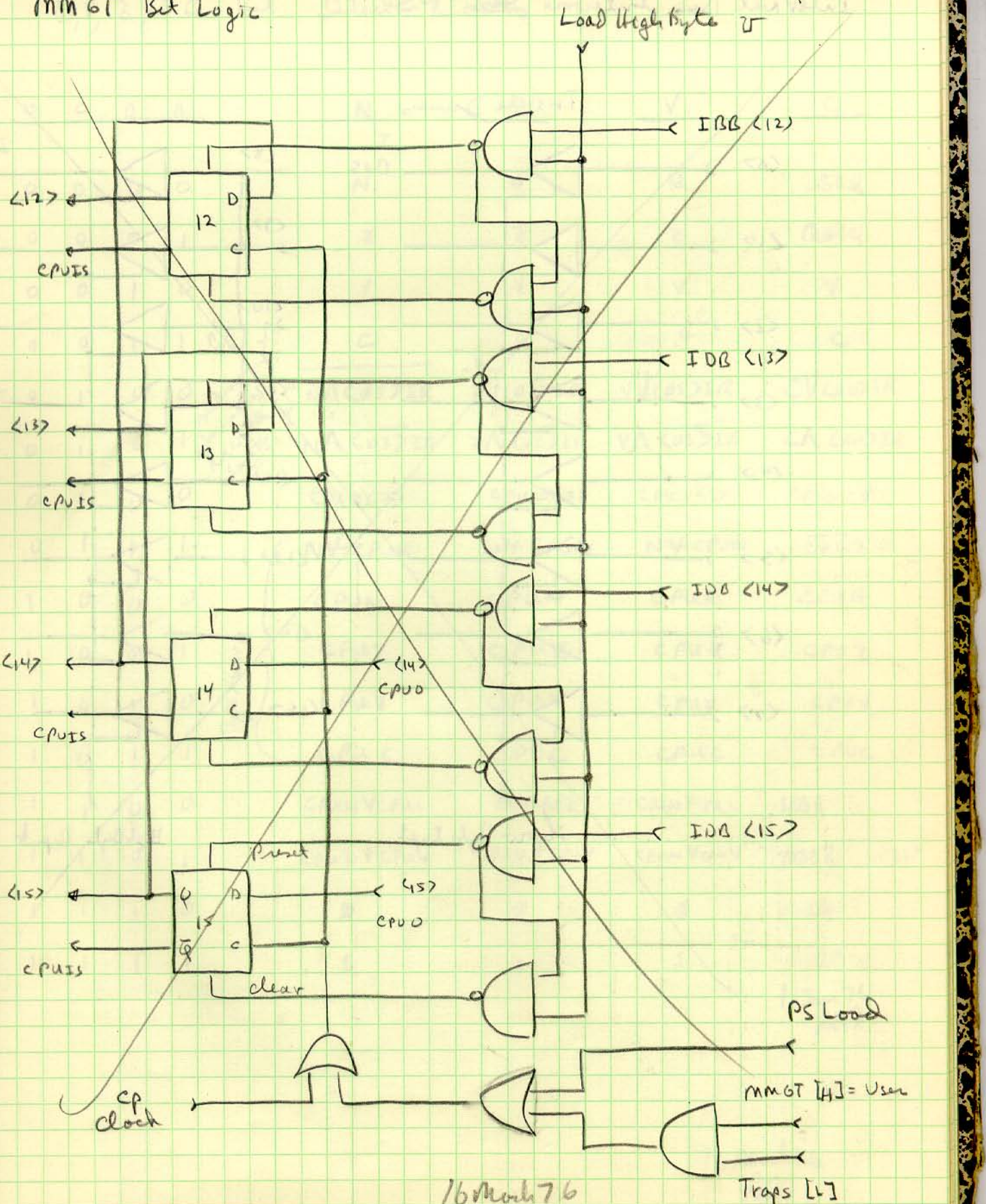
PS Bits	Kernel RTI/RTT	Explicit PS Address
<03:00>	from stack	*
<04>	from stack	cannot be changed
<07:05>	from stack	*
<13:12>	copied from <15:14>	*
<15:14>	from stack	*

* in Kernel may be accessed
in User may be changed if address is mapped to the User

16 March 76
BD

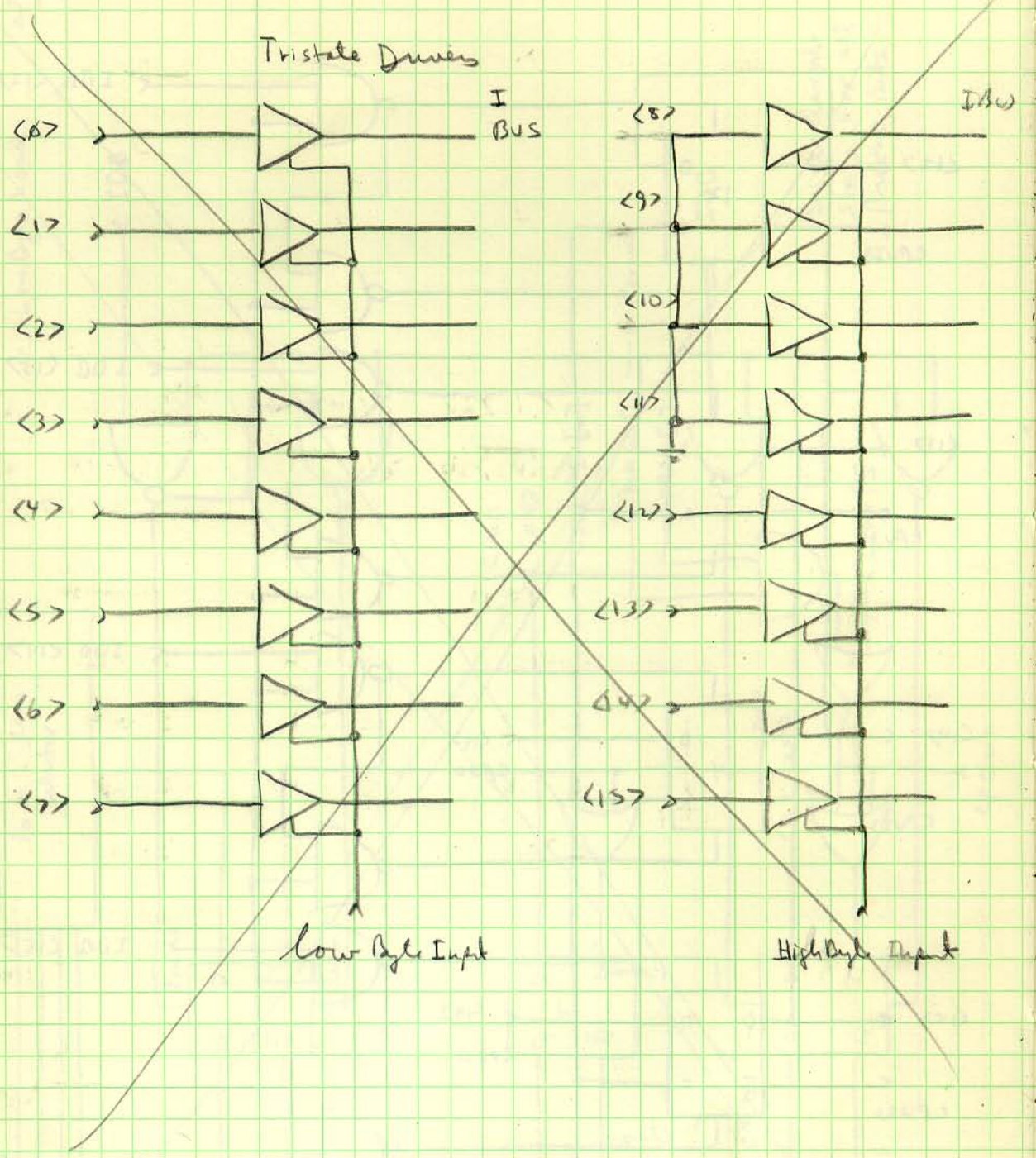

16 channels of Bit loading
total of 16 bits of Microcode
MCD select
16 to 1 line Selector
Low Byte Load
N Z V C Logic
Priority's
IDB
N, Z, V, C
Preset
D
Q
C
CPUIS
clear
CPUO
<04>
T
Load PS [L]
MMGT [L]
an Kernel
TRAPS[L]
IDB
<07:05>
CPUU
Load PS
Single decode stage for all Priority Bits
cp clock

MMGT Bit Logic



16 March 76
ABD

Internal Bus interface to PS



N, V, Z, & C Bit Selector Codes

D	C	B	A		N	Z	V	C
0	0	0	0		N	N	N	LSBQ
0	0	0	1		Z	Z	Z	MSBQ
0	0	1	0		V	V	V	V
0	0	1	1	B/W	C	C	C	C
0	1	0	0	Set	$\bar{N} \wedge \langle \bar{\emptyset 3} \rangle IR$	$Z \vee \langle \emptyset 2 \rangle IR$	$V \vee \langle \emptyset 1 \rangle IR$	$C \vee \langle \emptyset 0 \rangle IR$
0	1	0	1	clear	$N \wedge \langle \bar{\emptyset 3} \rangle IR$	$Z \wedge \langle \bar{\emptyset 2} \rangle IR$	$V \wedge \langle \bar{\emptyset 1} \rangle IR$	$C \wedge \langle \bar{\emptyset 0} \rangle IR$
0	1	1	0		CPU0<3>	CPU0<2>	CPU0<1>	CPU0<0>
0	1	1	1		N ∀ CPUN	N ∀ CPUN	N ∀ CPUN	$\overline{CPUZ}$
1	0	0	0		CPUN	CPUN	CPUN	CPUN
1	0	0	1		CPUZ	CPUZ	CPUZ	CPUZ
1	0	1	0		CPUV	CPUV	CPUV	CPUV
1	0	1	1		CPUC	CPUC	CPUC	CPUC
1	1	0	0		CPUN ∀ CPUC	Z ∧ CPUZ	CPUN ∀ CPUC	LSBR
1	1	0	1		CPUN ∀ CPUV	CPUN ∀ CPUV	CPUN ∀ CPUV	MSBR
1	1	1	0		∅	∅	∅	MSBA
1	1	1	1		1	1	1	MSBI

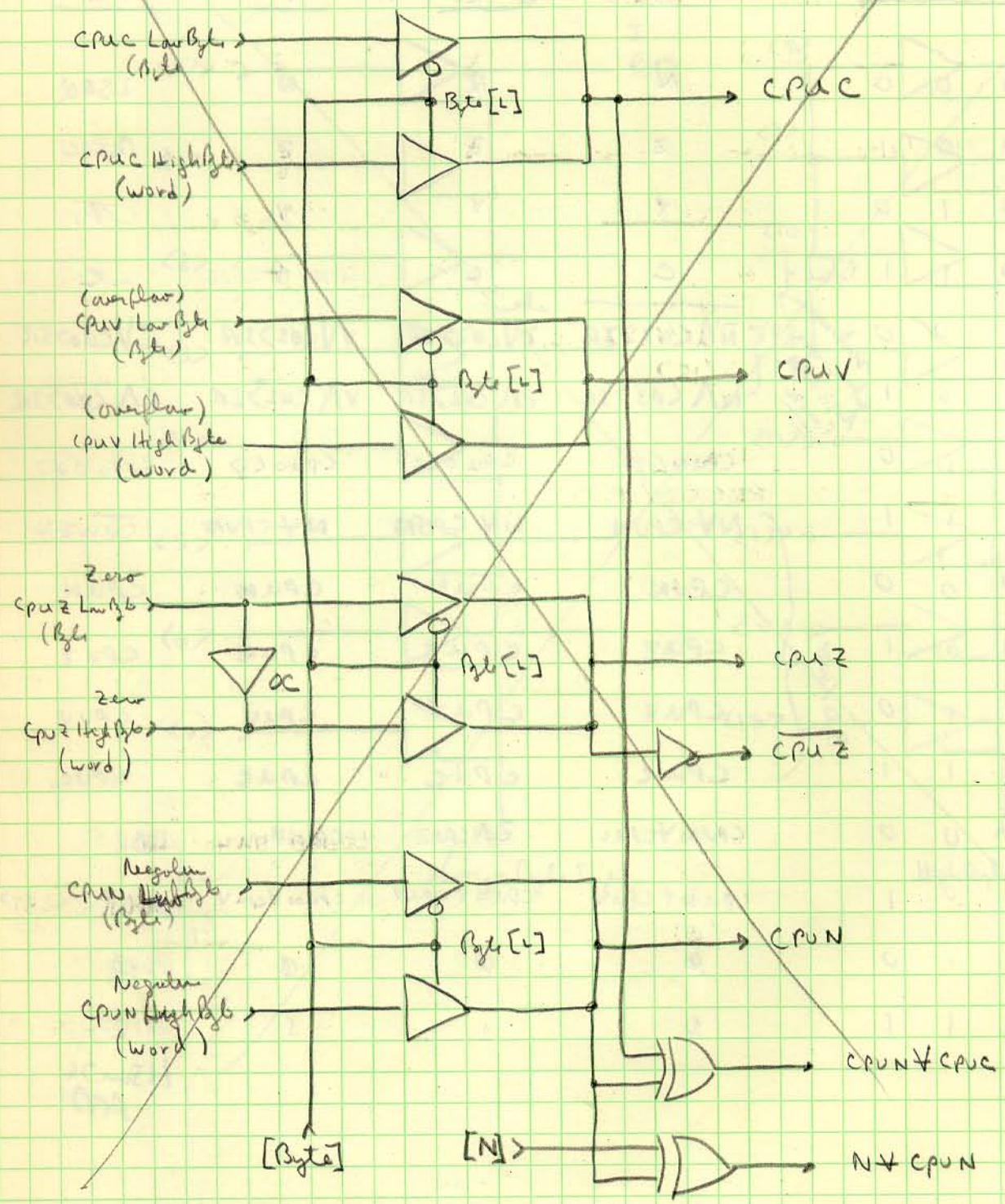
Byte/Word

Next two using Logic CPUs 6 June 76

11 June 76
AM

16 March 76
AM

Byte / word Logic for CPU N, Z, V, C



Shift/Rotate Control logic

8 Bits of Microcode Used for Control

2 Bits of Microcode Used for Byte Force/Enable

Shift data Codes (all from shift outputs of MCP)

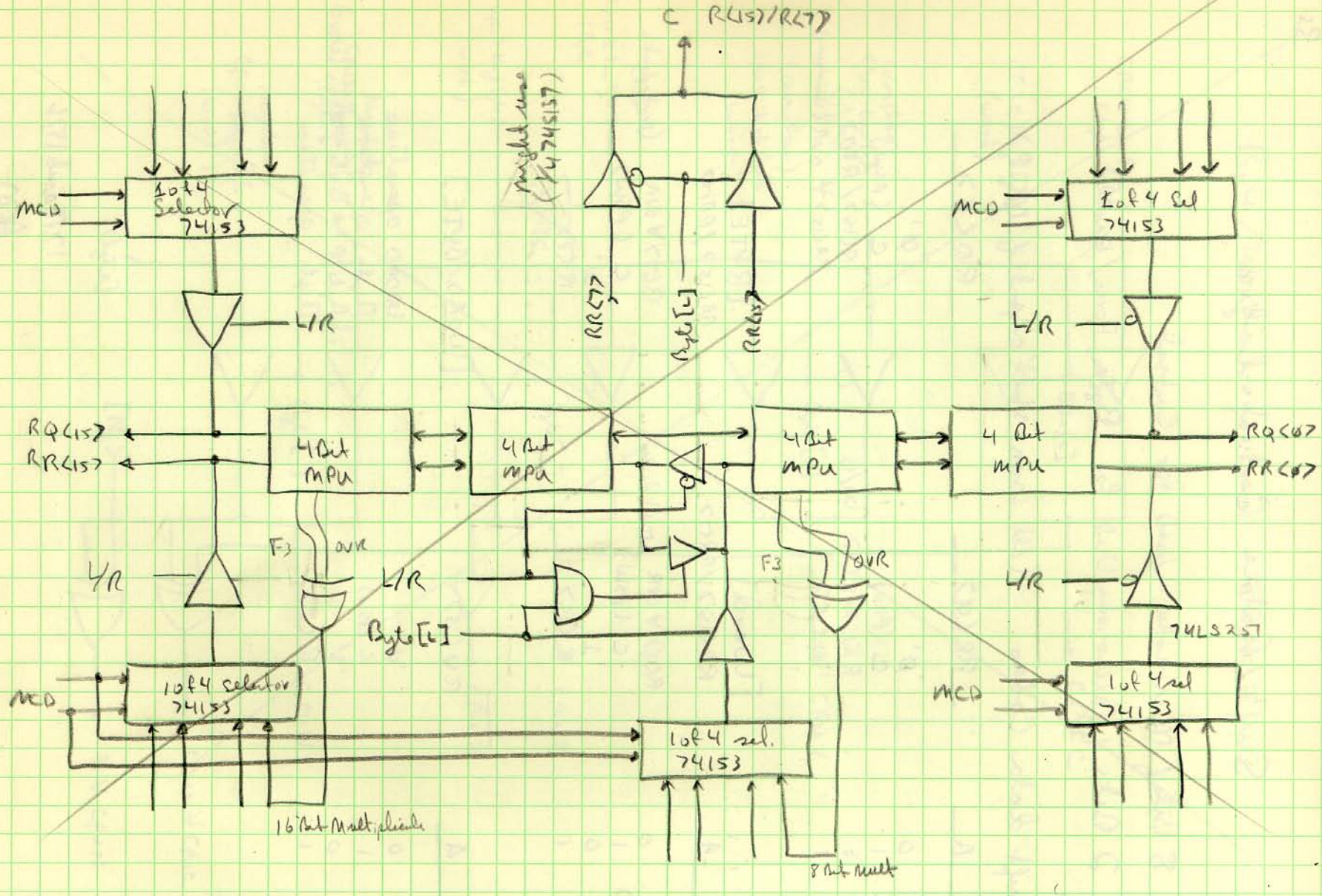
B	A	RR<Ø>	RQ<Ø>
0	0	'Ø'	'Ø'
0	1	C (PSW)	C (PSW)
1	0	R<15>/R<7> W/B	R<15>/R<7>
1	1	Q<15>	RR<15> ∀ OVR (Division)

B	A	[Word] RR<15>/RR<7>	[BYTE] RR<15>/RR<7>
0	0	R<15> ∀ OVR (multiplication	R<7> ∀ OVR (multiplication
0	1	C (PSW)	C (PSW)
1	0	'1'	'1'
1	1	R<15>	R<7>

B	A	RQ<15>	[WORD/BYTE]
0	0	'Ø'	WORD operations
0	1	C (PSW)	Byte operations
1	0	'1'	if Byte OP → Byte; else Word
1	1	R<Ø7>	Byte operation

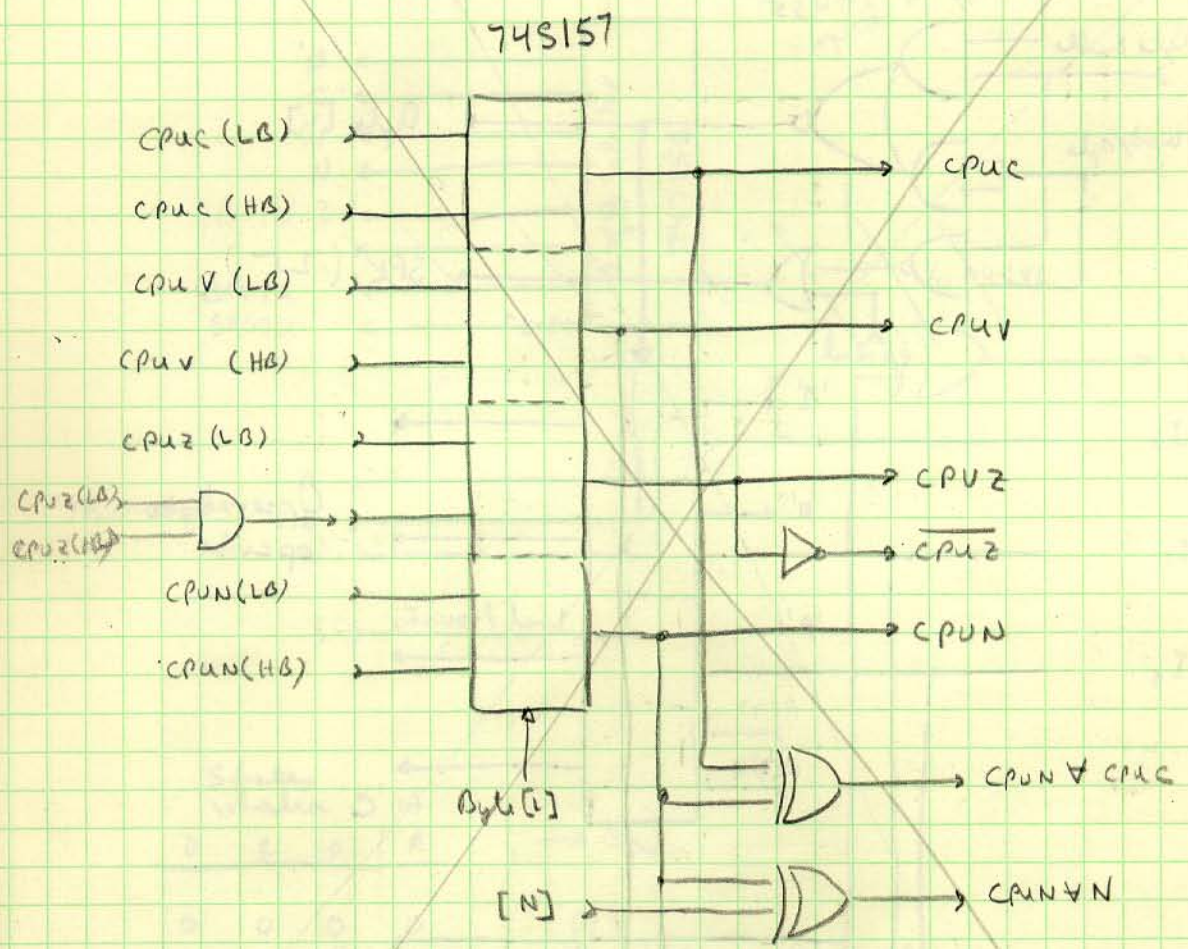
17 March 1976

ABO



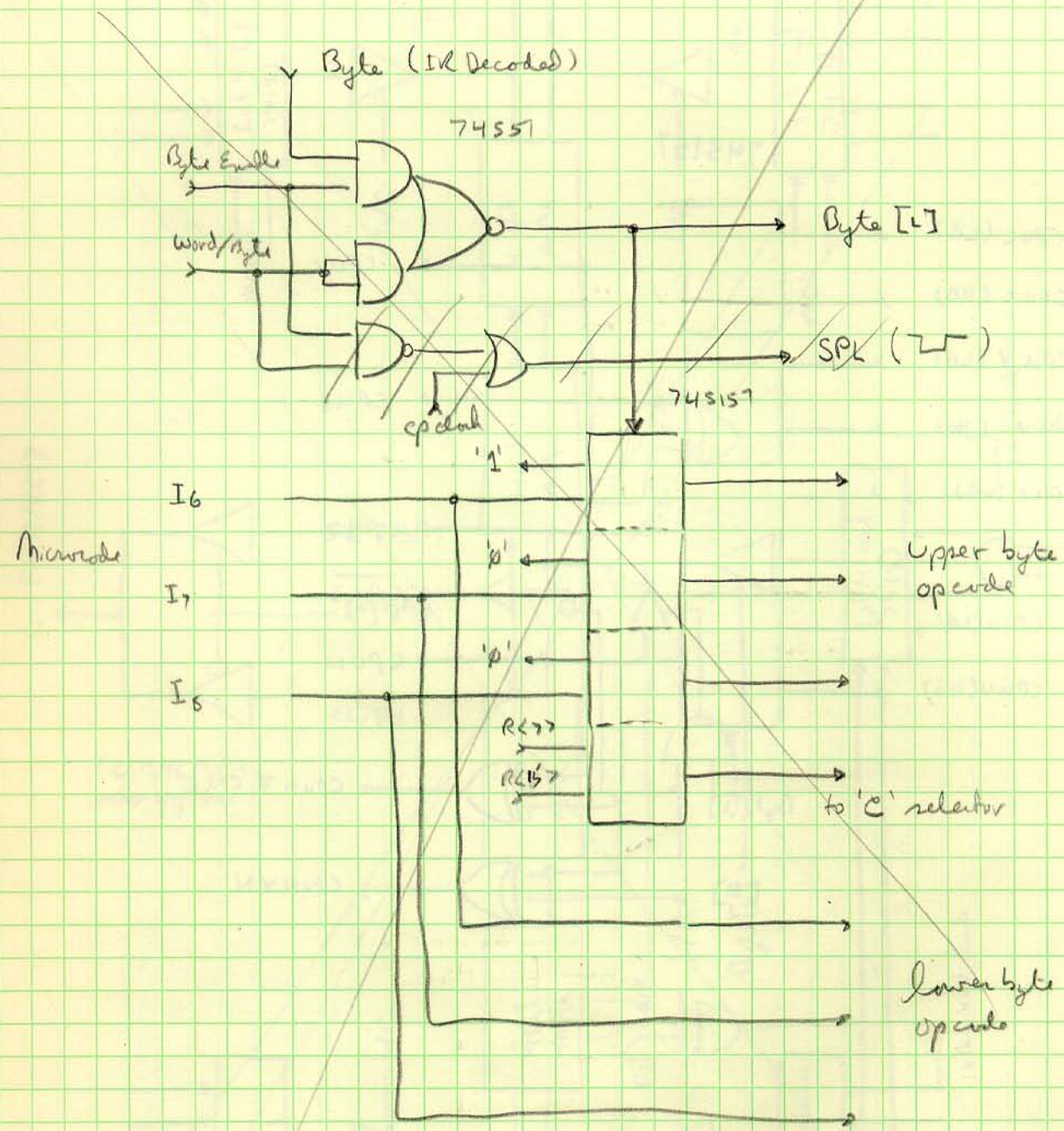
L/R ie '0' for left '1' for right

Alternate Word/Byte Logic for CPUN, Z, V, C

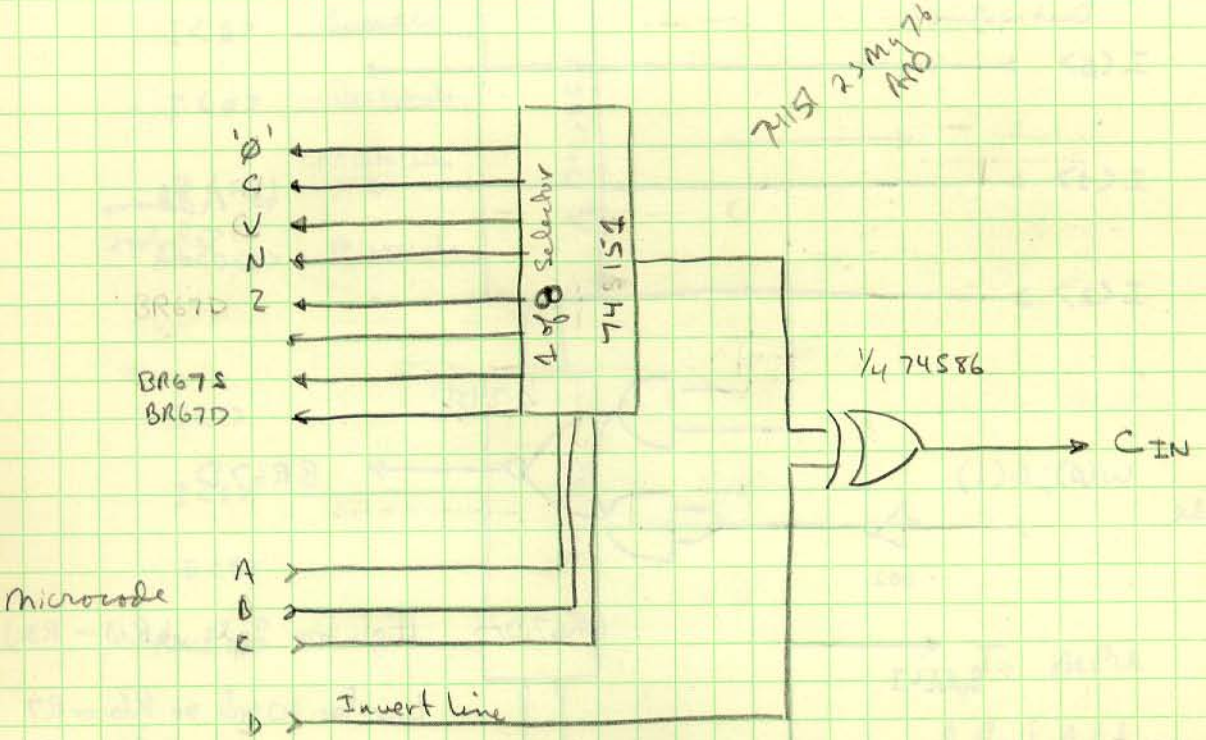


18 March 76
ARP

Byte Enable logic for MPU



Carry IN Logic for the MPU

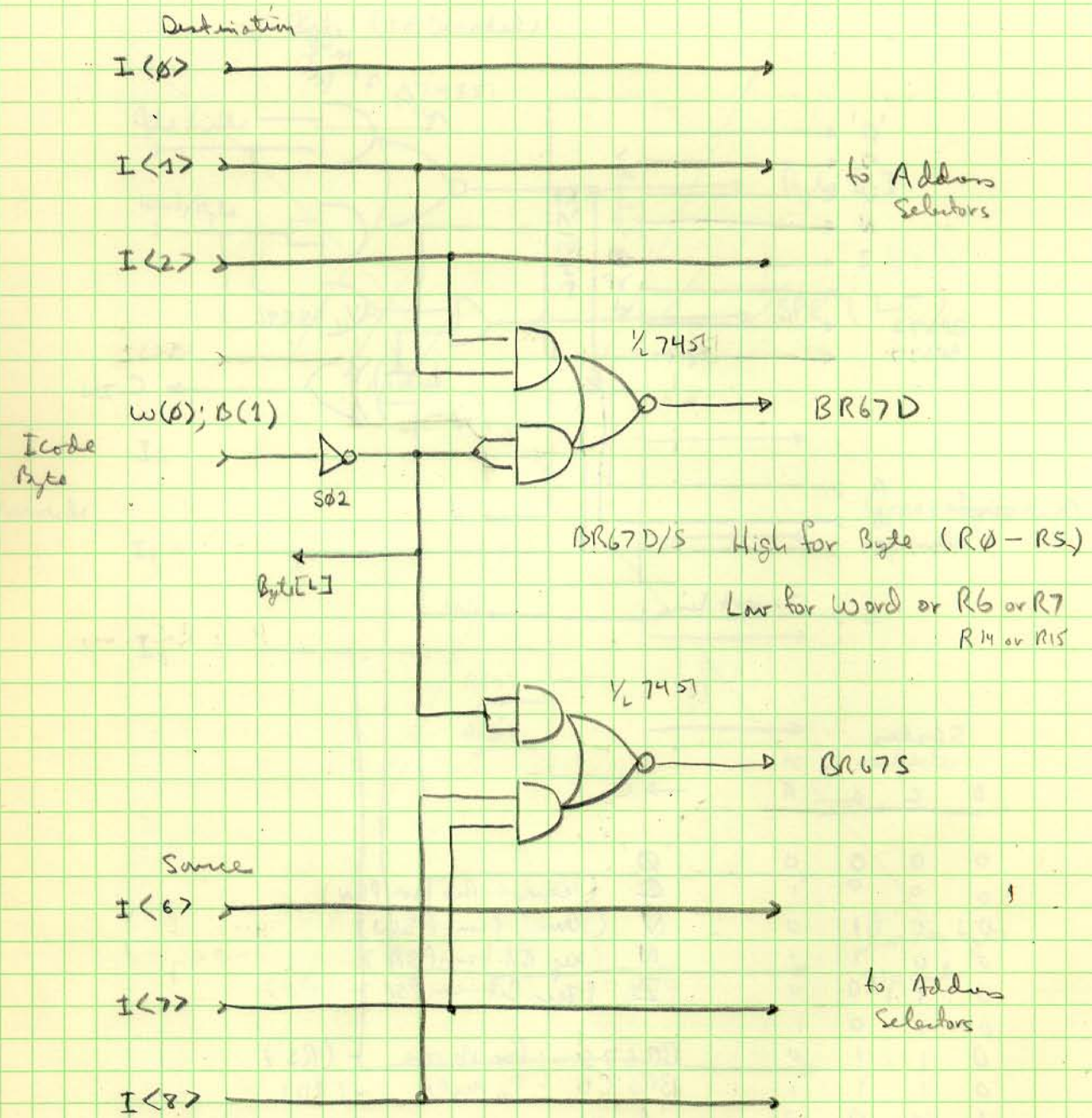


States

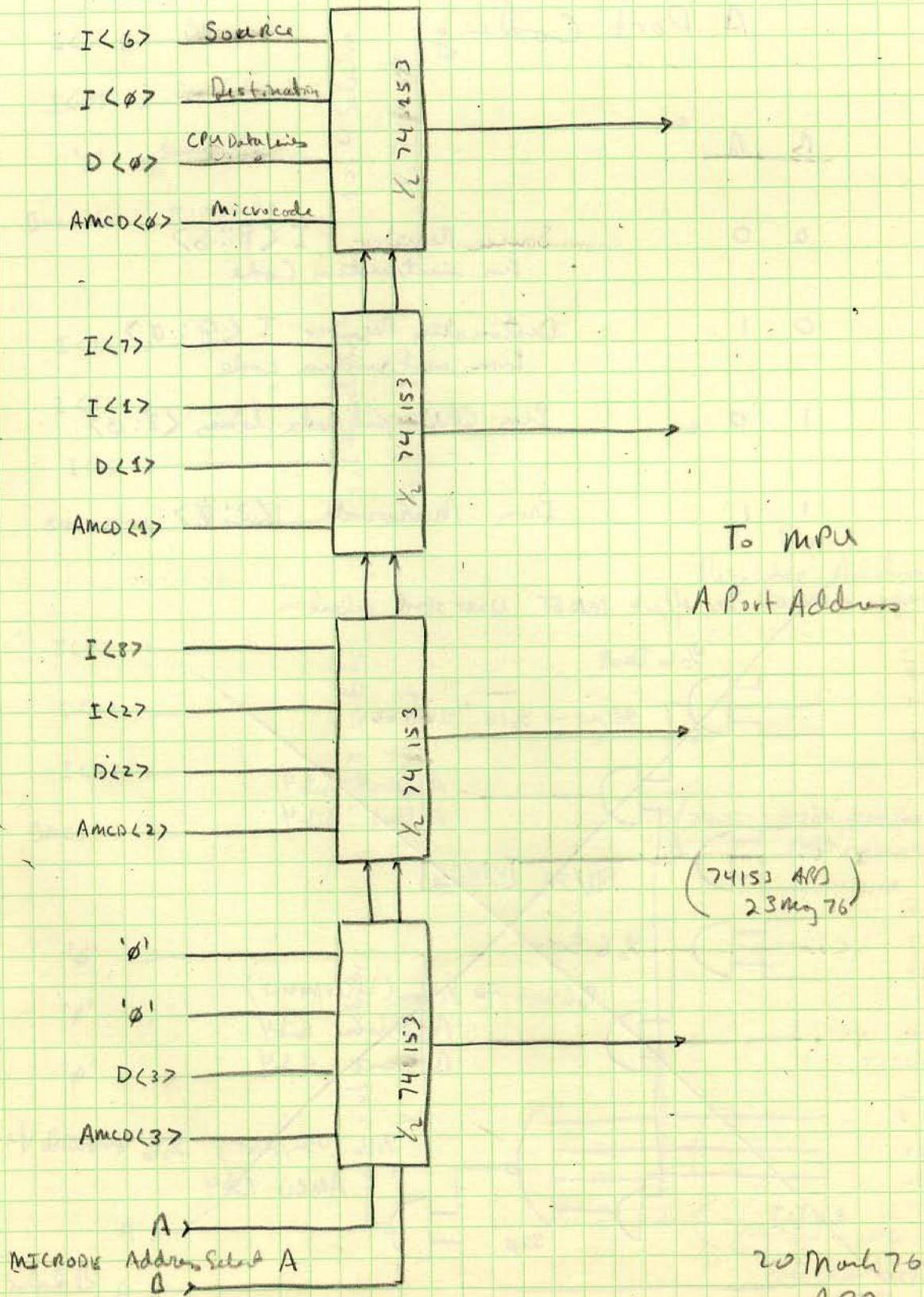
D	C	B	A	→ C_{IN}
0	0	0	0	'Ø'
0	0	0	1	C (Carry Bit from PSW)
0	0	1	0	V (Ovn from PSW)
0	0	1	1	N (Neg Bit from PSR)
0	1	0	0	Z (Zero Bit from PSR)
0	1	0	1	
0	1	1	0	BR67S ; for Mode -(RS)
0	1	1	1	BR67D ; for Mode -(RD)
1	0	0	0	'1'
1	0	0	1	$\overline{C}$ (inverted carry from PSW)
1	0	1	0	$\overline{V}$ (inverted Ovn from PSW)
1	0	1	1	$\overline{N}$ (" Neg from PSR)
1	1	0	0	$\overline{Z}$ (" Zero from PSR)
1	1	0	1	
1	1	1	0	$\overline{BR67S}$; for Mode (RS)+
1	1	1	1	$\overline{BR67D}$; for Mode (RD)+

20 March 76
ARD

Generation of BR67 D/S



A Port Address Selector

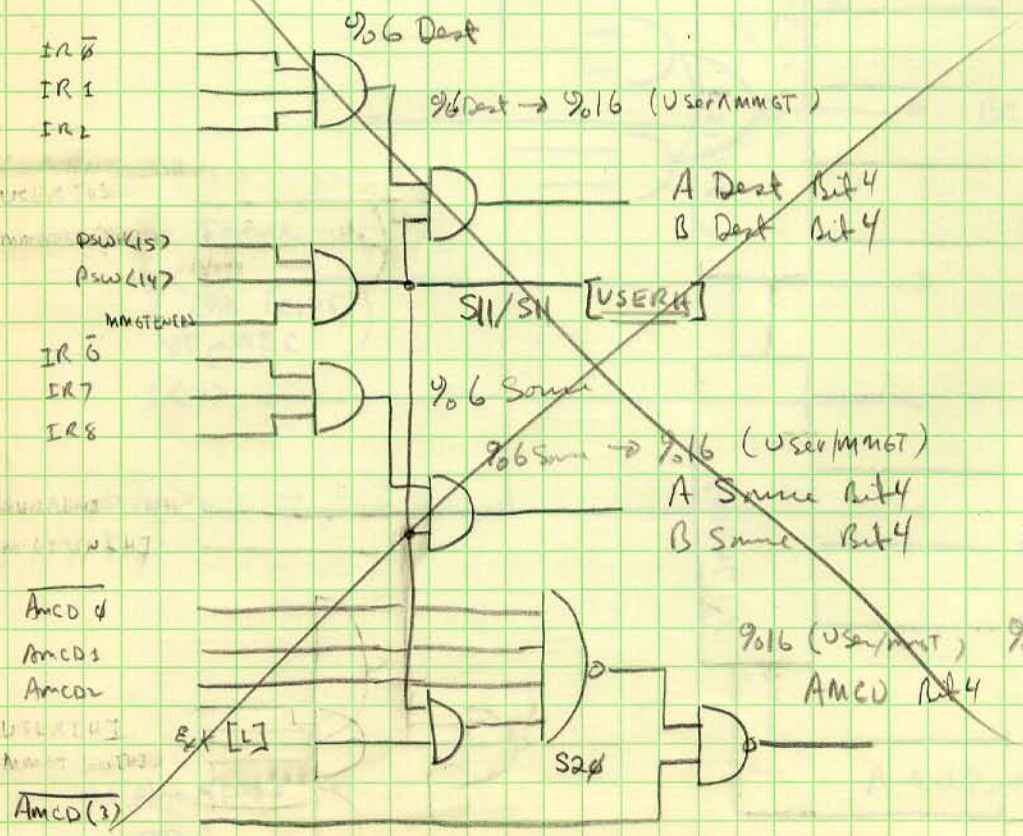


20 March 76
APD

A Port Coding

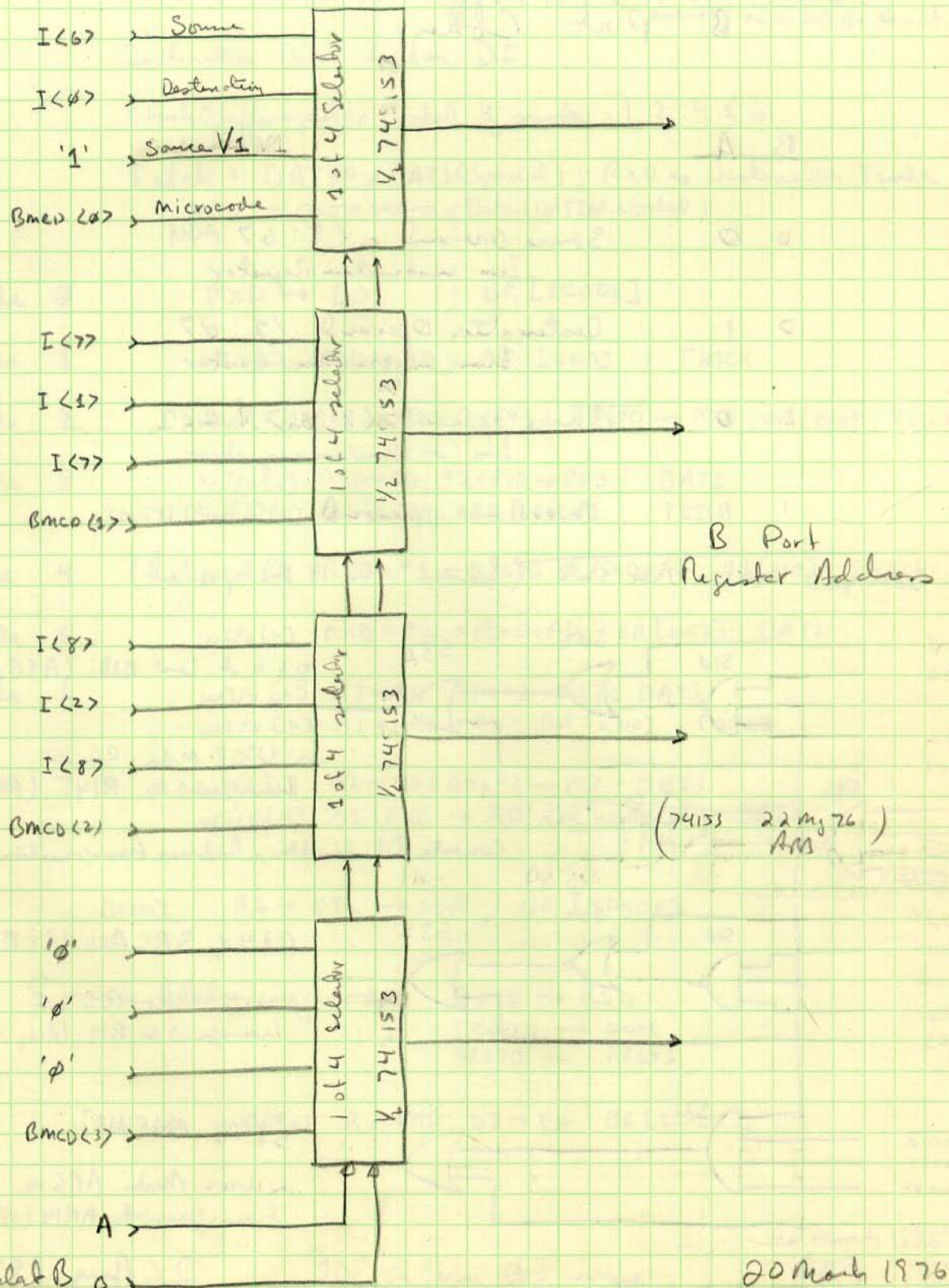
B	A	
0	0	Source Register I<8:6> from instruction Code
0	1	Destination Register I<2:0> from instruction code
1	0	From CPU Input Data lines <3:0>
1	1	From Microcode <3:0>

Changes needed to effect MMGT user stack selection



Microco
Addr

B Port Address Selector

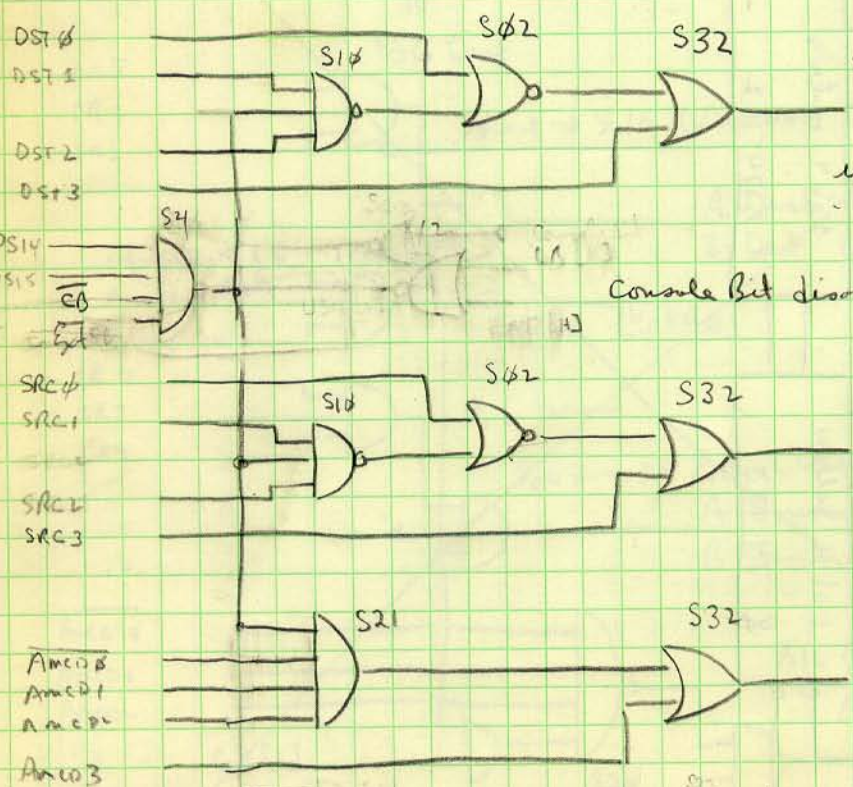


20 March 1976
ASD

B Port Coding

B	A	
0	0	Source Operand <8:6> from instruction Register
0	1	Destination Operand <2:0> from instruction register
1	0	Source Operand <8:06> V 001 from instruction register
1	1	Microcode operand <3:6>

Redesigned MM6T User stack (former with 4 bit Ry select)



Bit 4 of Dest Add (A&B)

in USER Mode R6 is transformed to R14 ($R16_8$)

Console Bit disables Register Address transformation

Bit 4 of SRC ADD (A&B)

in User Mode R6 is transformed to R14 ($R16_8$)

Bit 4 of AMCD Add,

in User Mode AR6 is transformed to AR14 ($AR16_8$)

extra S32 used in 'SOR'

26 Aug 1976

BRD

12 words Refined Destination Addressing Code

Destination ADDress left in AD (except Mode Ø)
                                 caused by extended Arithmetic

Destination Data left in DI

stack Overflow register (SOR) loaded by mode 1, 2, 4 & 6

Fetch = DATIP / DATI(special) ; RXD is Destination Icode
  NAD — no access of Data (DATI/P aborted)
  NDA

Mode Ø RXD → DO ; BR [ICODE]

Mode 1 WI/O[L] RXD → AD ; BR [END] ; FETCH

mode 2 WI/O[L] RXD → AD ; RX + #1 + $\overline{BR67D}$ → RXD ; BR [END] ; FETCH

mode 3 WI/O[L] RXD → AD ; #2 + RXD → RXD ; DATI
[OUTI] WI/O [H] DI → AD ; BR [ICODE] ; FETCH

mode 4 WI/O [L] RXD − #1 − BR67D → RXD + AD ; BR [END] ; FETCH

mode 5 WI/O [L] RXD − #2 → RXD + AD ; BR [OUTI] ; DATI

mode 6 WI/O [L] R7 → AD ; R7 + #2 → R7 ; DATI
  WI/O [H] DI + RXD → AD ; BR [END] FETCH

mode 7 WI/O [L] R7 → AD ; R7 + #2 → R7 ; DATI
  WI/O [H] DI + RXD → AD ; BR [OUTI] ; DATI

[END] R6 − STL → SOR ; BR [ICODE]

For Some Addressing code RXD → RSD
  [CODE → ENDI
  BR67D → BR67S

[ENDI] WI/O[H] R7 → AD ; DI → T Ø ; BR [ICODE]

20 March 1976
ARD

CPU Microcontroller Logic

The microcontroller sequences the CPU through the Stored Microprogram by controlling Jmps, Conditional Branches, JSR's, and Returns.

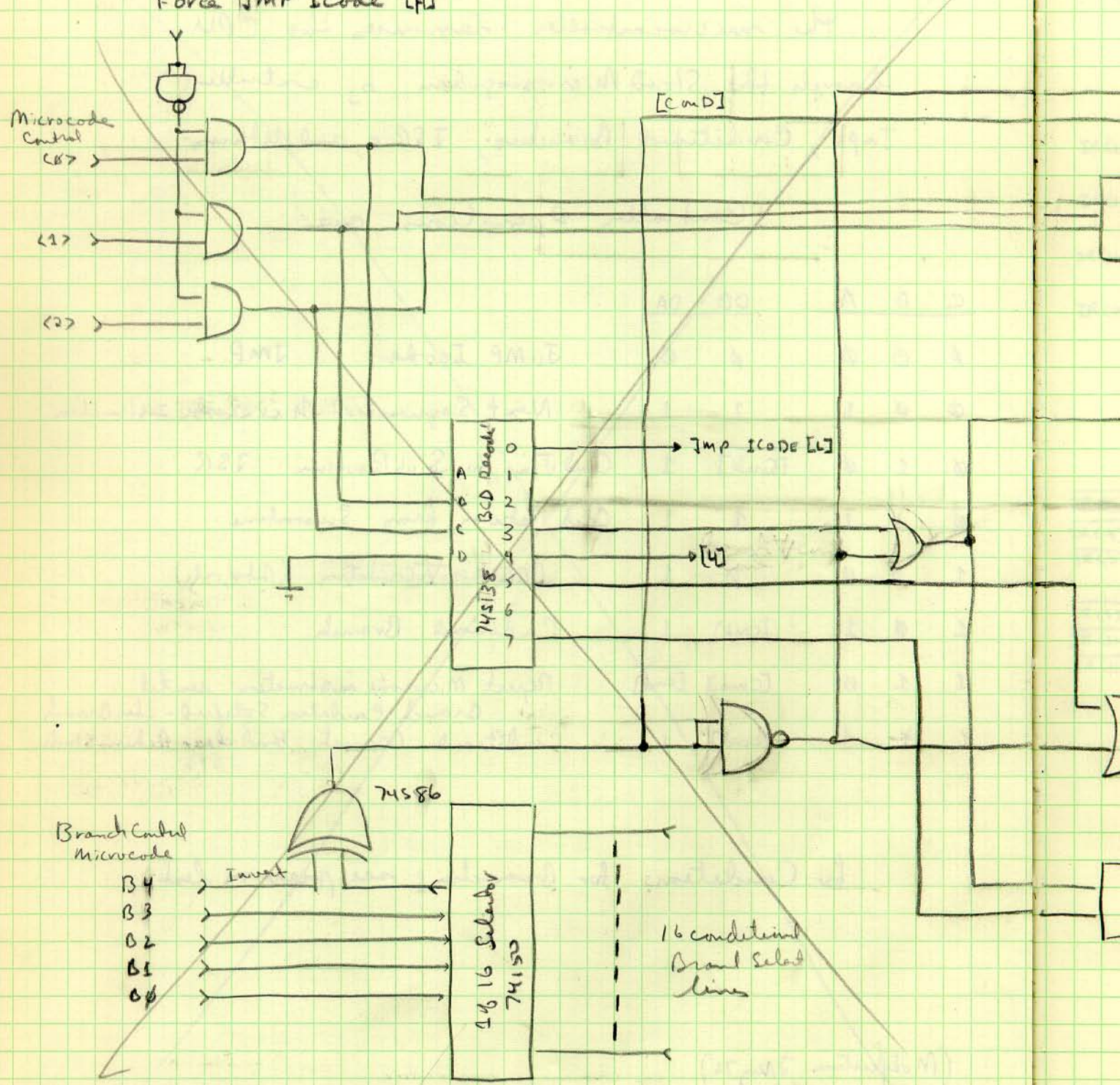
Controller Operations are

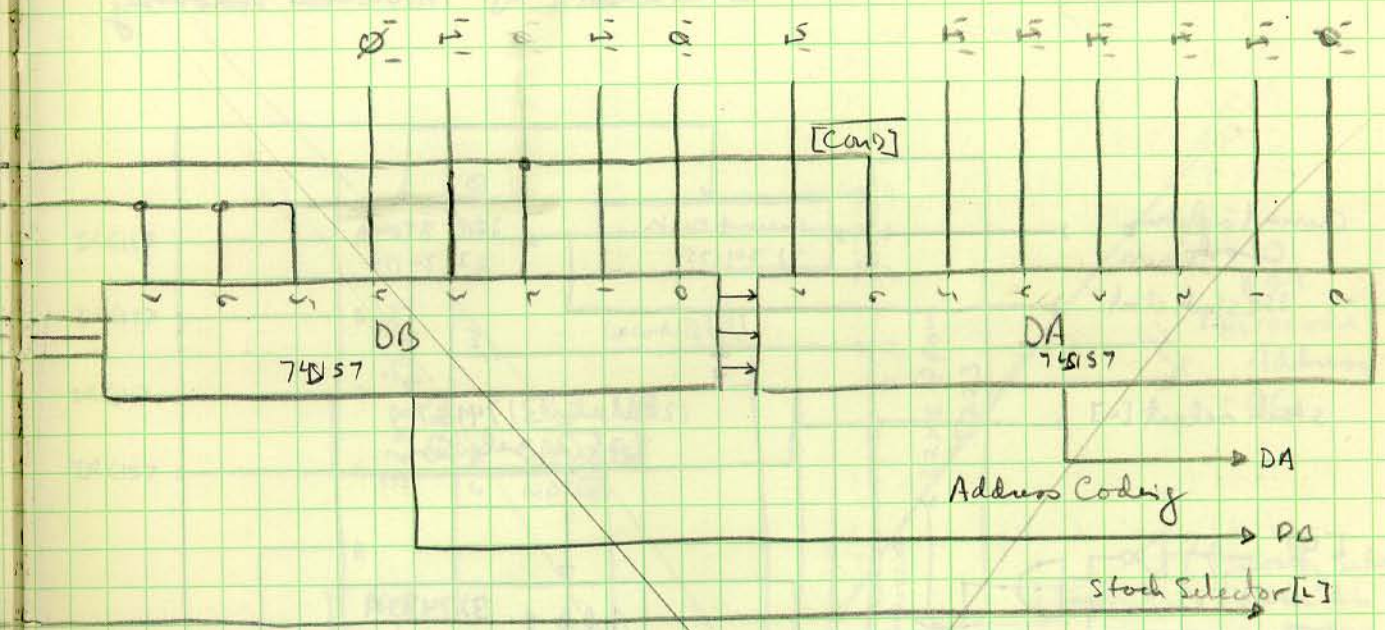
C	D	A	DB	DA	
0	0	0	0	0	JUMP Icode JMP
0	0	1	1	1	Next Sequential Microcode Instruction
0	1	0	$\overline{[Cond]}$	1	Cond Jump to SubRoutine JSR
0	1	1	1	1	Cond Return from Subroutine
1	0	0	0	1	BRANCH ∀ Condition (Bit 0 only)
1	0	1	[cond]	1	Conditional Branch
1	1	0	[Cond]	$\overline{[Cond]}$	Repeat Microcode instruction until Branch Condition Satisfied - then Branch
1	1	1	[cond]	1	Conditional Branch and clear Return Stack

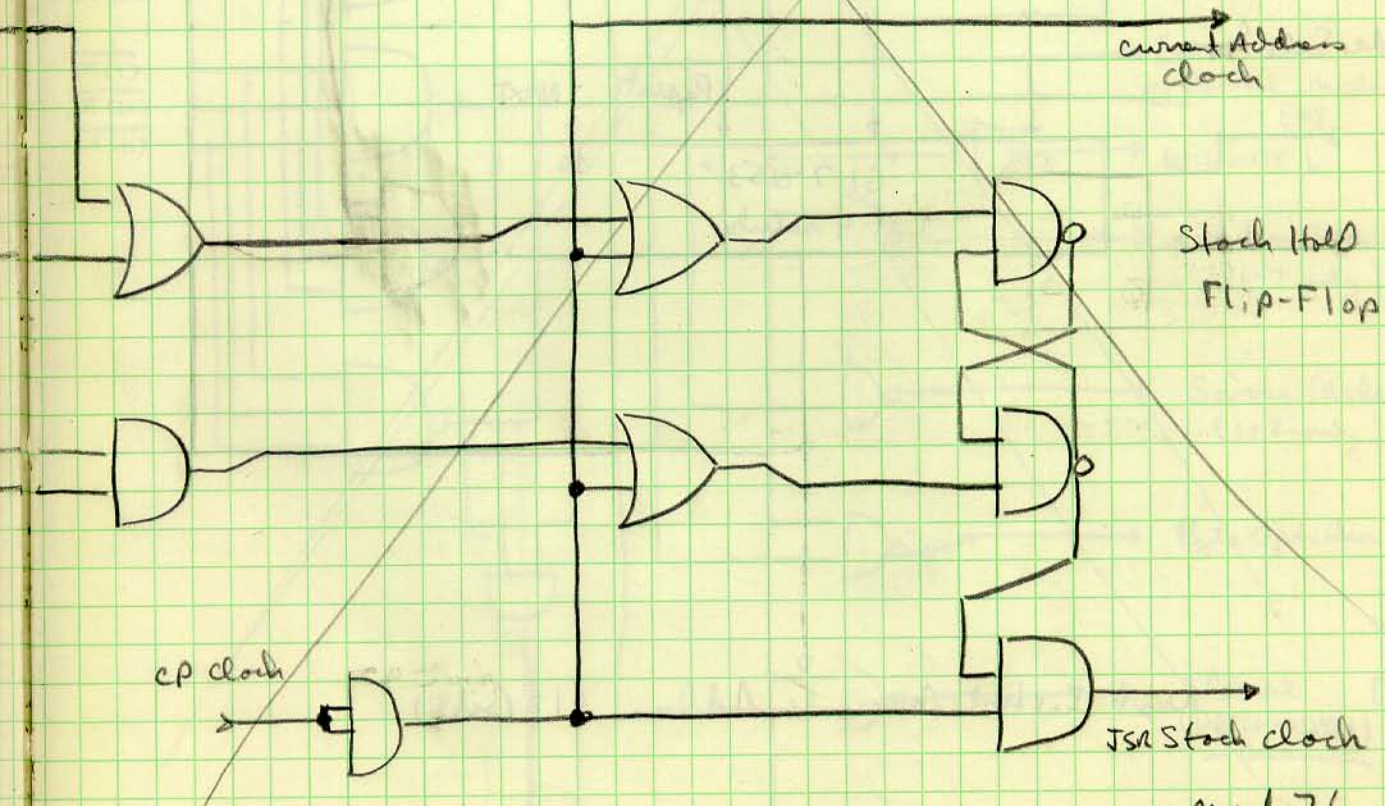
for Conditions for Branching see previous list.

(Modification 2 May 76 ARB)

21 March 1976
ARB

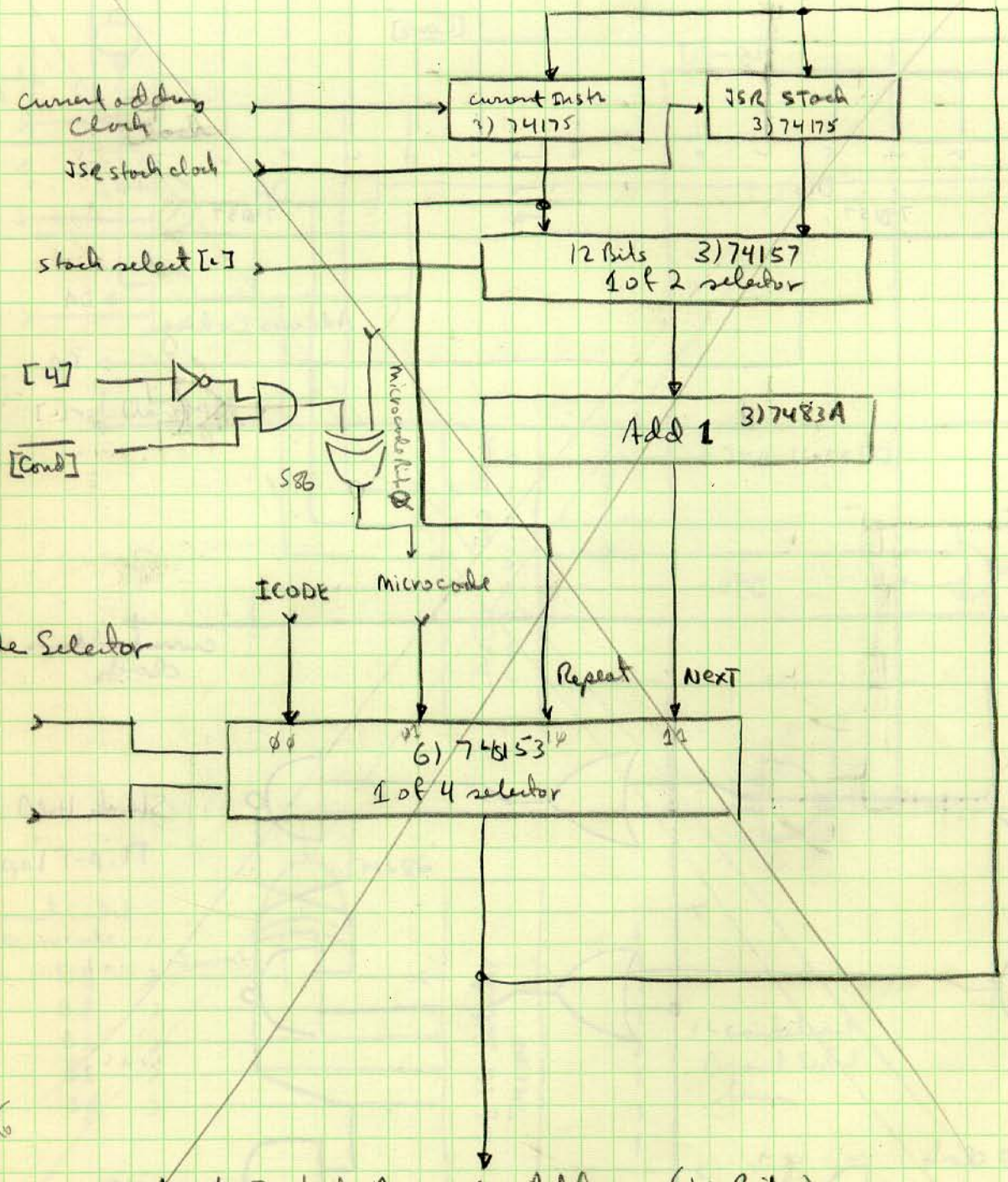

Force JMP Icode [H]
Microcode Control
<7>
<1>
<2>
[CMD]
74S138
BCD Decode
A
B
C
D
0
1
2
3
4
5
6
7
JMP ICODE [L]
[H]
74S86
Branch Control Microcode
B4
B3
B2
B1
B0
Invert
16:16 Selector
74153
16 conditional Branch Select lines



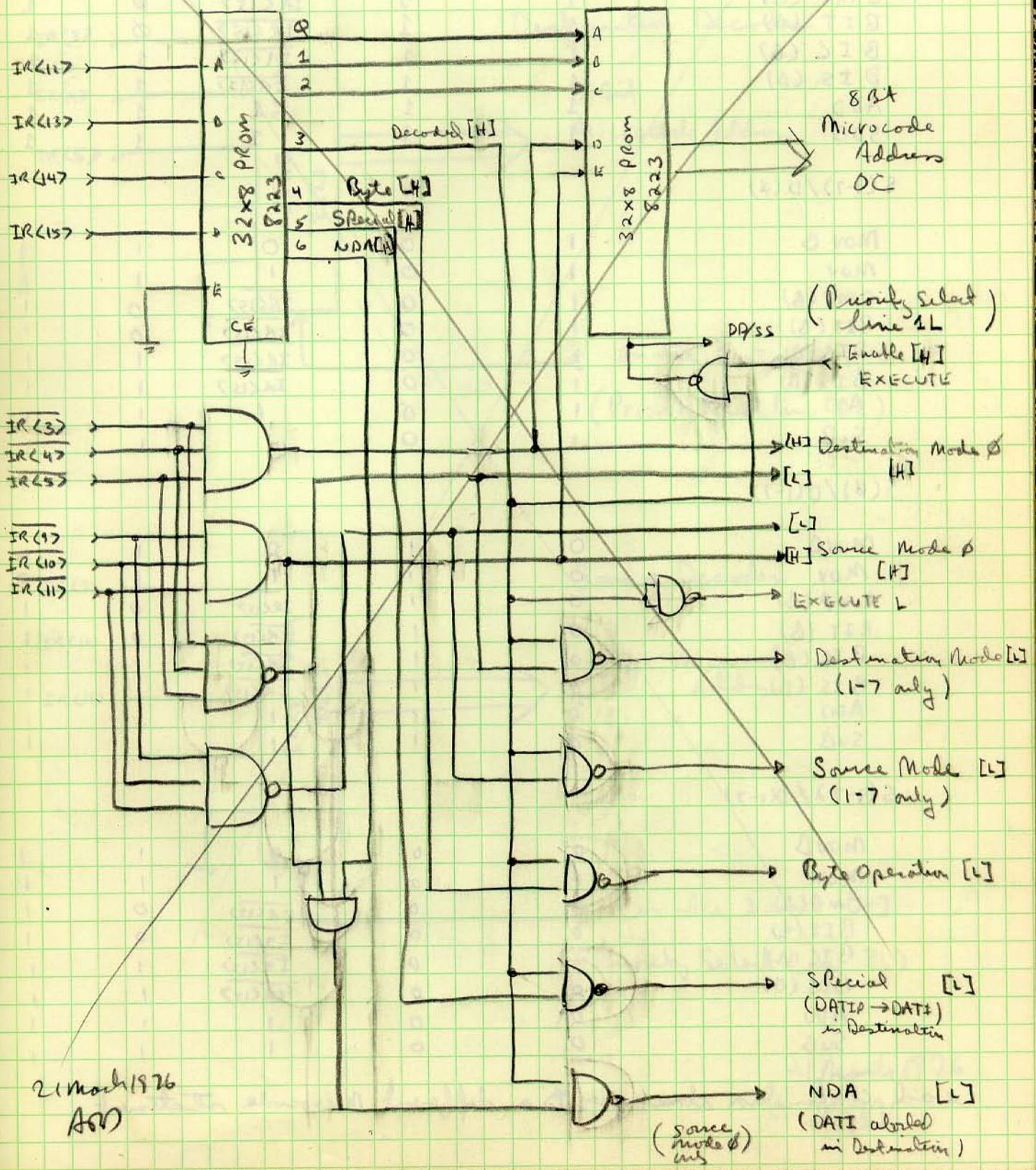


21 March 76
AMB

12 Bits of Microcode Address



Double Operand Instruction Register Decoder



2 March 1976
AGM

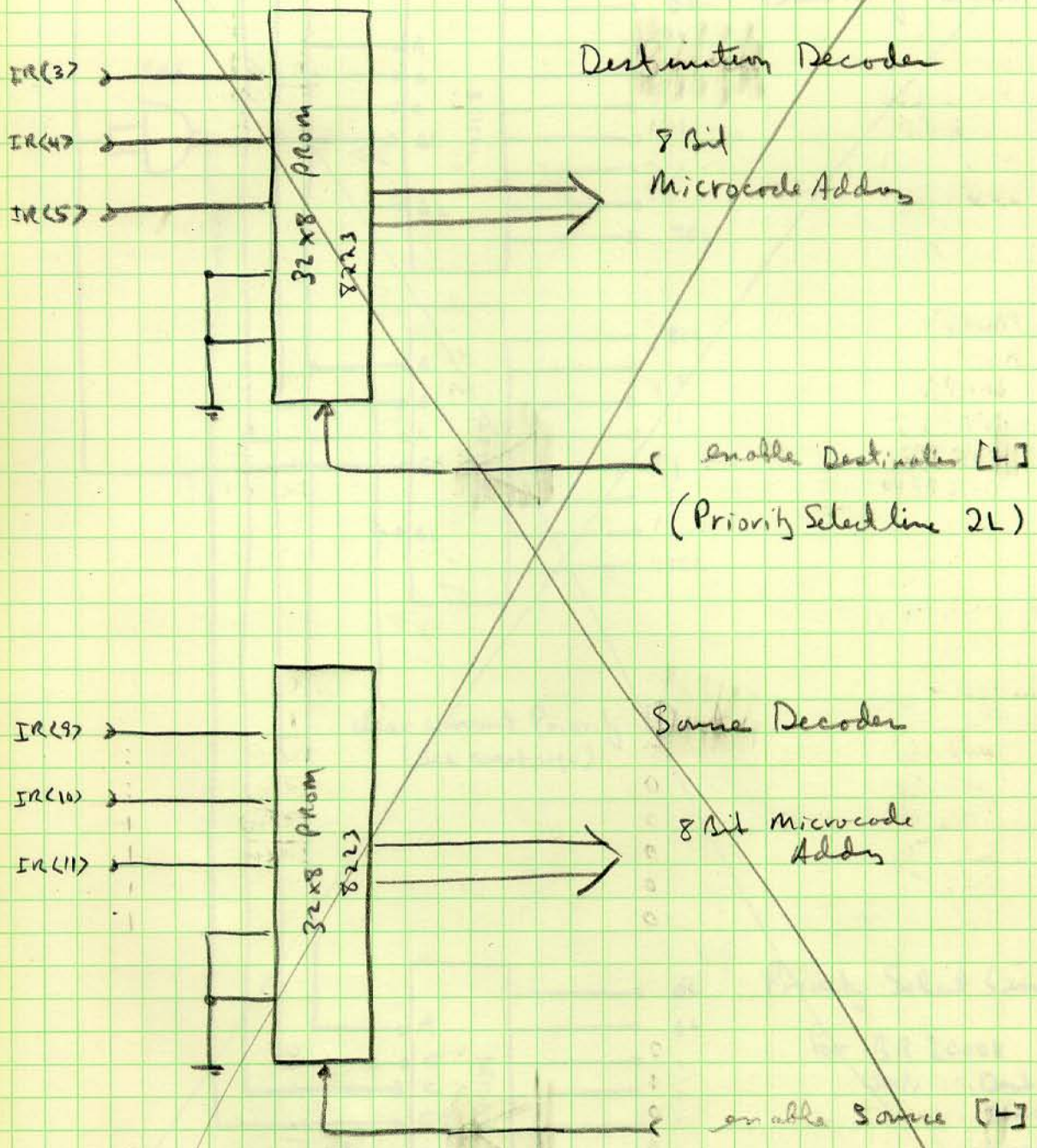
Decoded Signals for Dubb Operand

S(Ø) / D(Ø)	Dest Mode	Source Mode	Byte	SPCL (Applicable only for D(1-7))	NDA & S(Ø)
MOV B	1	1	Ø	1	Ø
MOV	1	1	1	1	Ø
CMP (B)	1	1	IR<15>	Ø	1
BIT (B)	1	1	IR<15>	Ø	1
BIC (B)	1	1	IR<15>	1	1
BIS (B)	1	1	IR<15>	1	1
ADD	1	1	1	1	1
SUB	1	1	1	1	1
S(1-7) / D(Ø)					
MOV B	1	0	0	1	1
MOV	1	0	1	1	1
CMP (B)	1	0	IR<15>	0	1
BIT (B)	1	0	IR<15>	0	1
BIC (B)	1	0	IR<15>	1	1
BIS (B)	1	0	IR<15>	1	1
ADD	1	0	1	1	1
SUB	1	0	1	1	1
S(Ø) / D(1-7)					
MOV B	0	1	0	1	0
MOV	0	1	1	1	0
CMP (B)	0	1	IR<15>	0	1
BIT (B)	0	1	IR<15>	0	1
BIC (B)	0	1	IR<15>	1	1
BIS (B)	0	1	IR<15>	1	1
ADD	0	1	1	1	1
SUB	0	1	1	1	1
S(1-7) / D(1-7)					
MOV B	0	0	0	1	1
MOV	0	0	1	1	1
CMP (B)	0	0	IR<15>	0	1
BIT (B)	0	0	IR<15>	0	1
BIC (B)	0	0	IR<15>	1	1
BIS (B)	0	0	IR<15>	1	1
ADD	0	0	1	1	1
SUB	0	0	1	1	1

each confeniation above goes to a different Microcode statement

Source / Destination Decoders

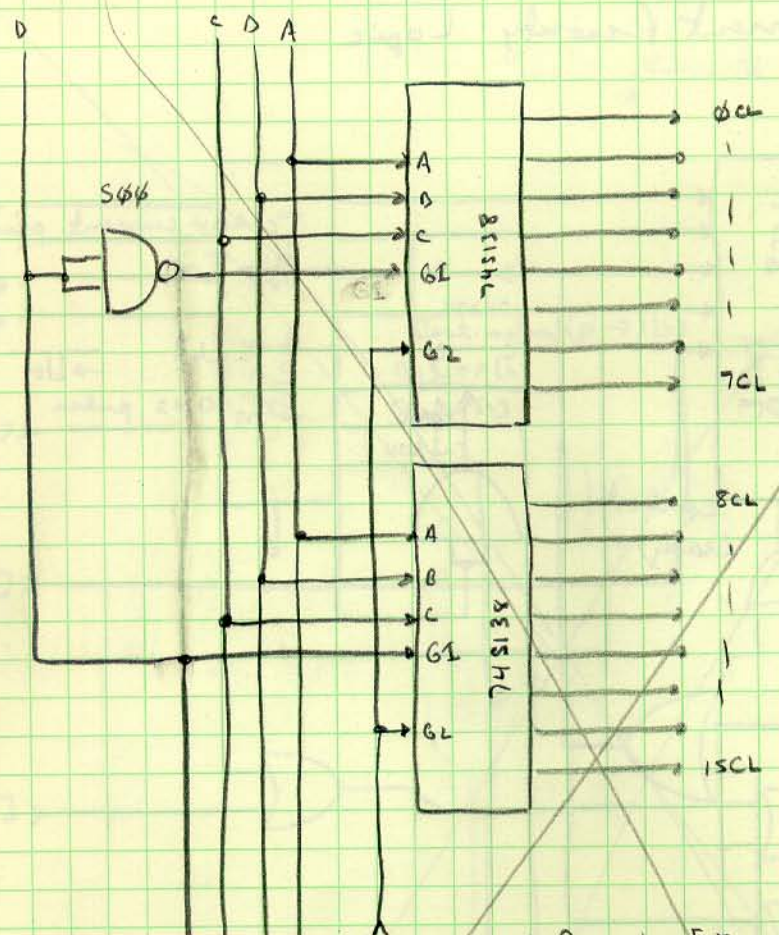
Each Source/Destination Mode points to a different Microcode Address

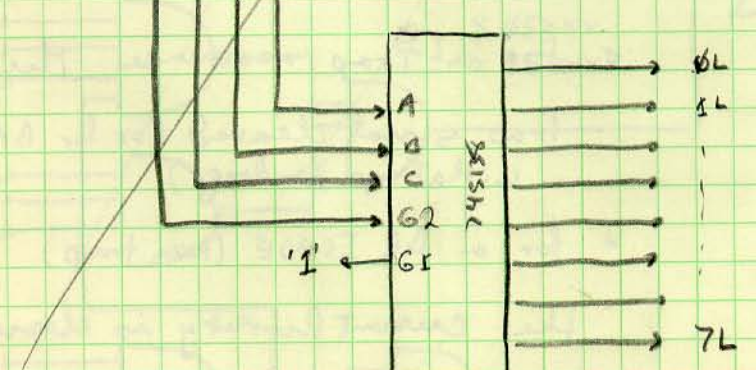


21 March 1976

ABD

Modification to the Priority Clear/Select Logic





Priority Clear lines

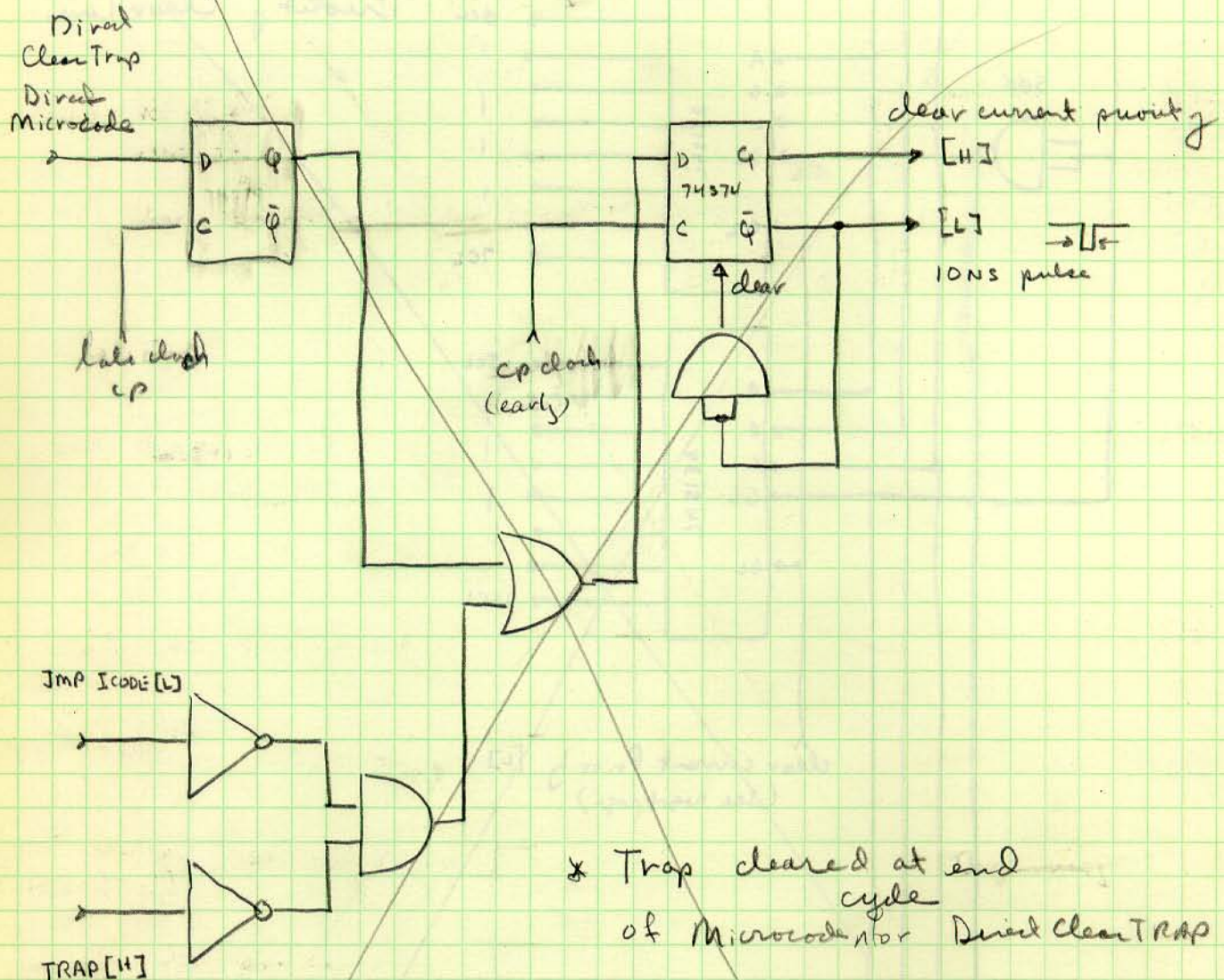
Priority Select lines

for BR ICODE (JMP)

Dest	2L
Source	3L
Execute	1L

23 March 1976
ARD

Clear Current Priority Logic

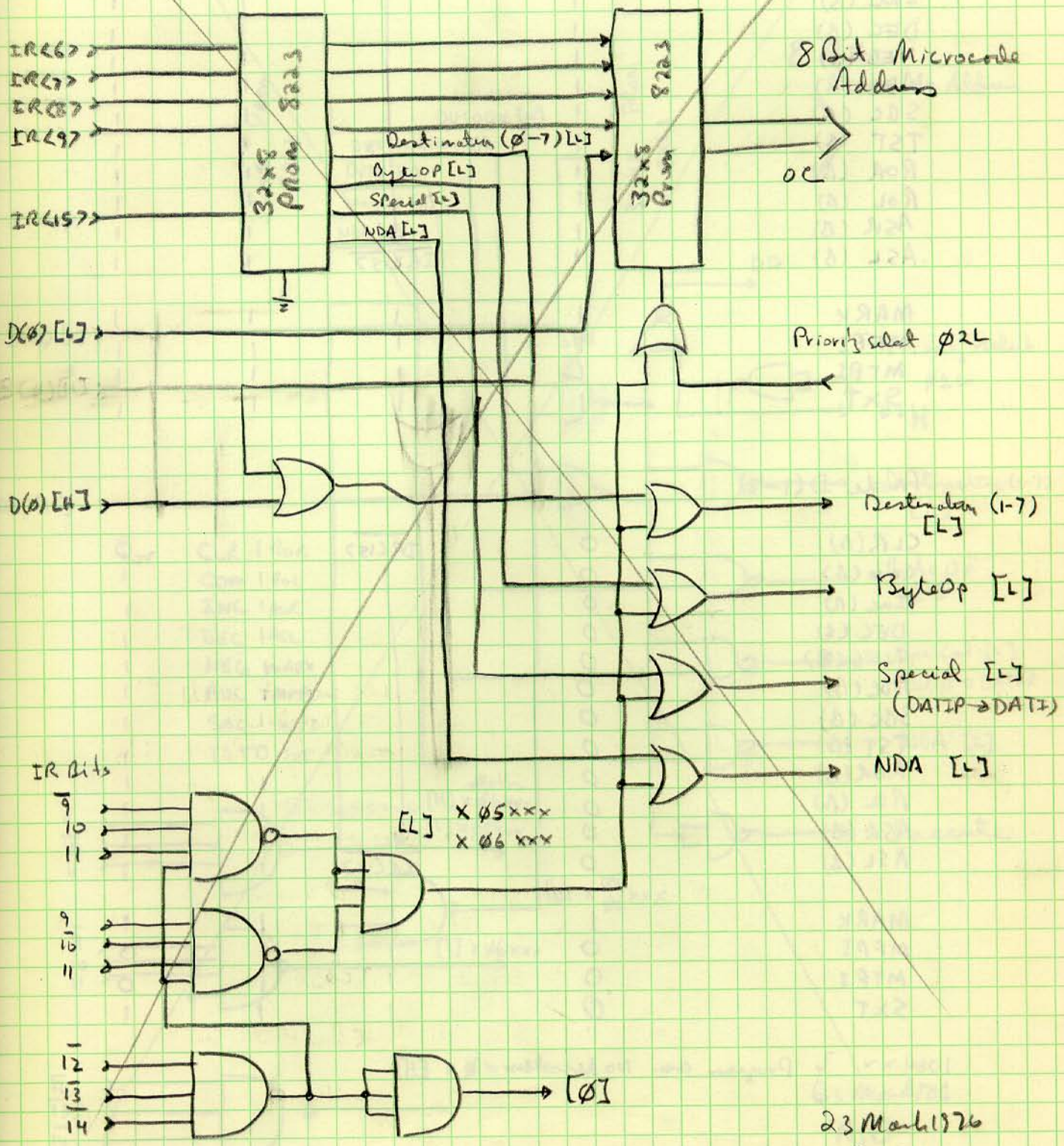


* Trap cleared at end of Microcode cycle for Direct Clear TRAP

* for a Trap occurrence the Trap is not cleared (or for BR Icode with a Trap pending)

* for a BR ICODE (Non trap) the current Priority is cleared at end of current microcycle

Single Operand Decoder



23 March 1976

AMB

Single Operand Coding

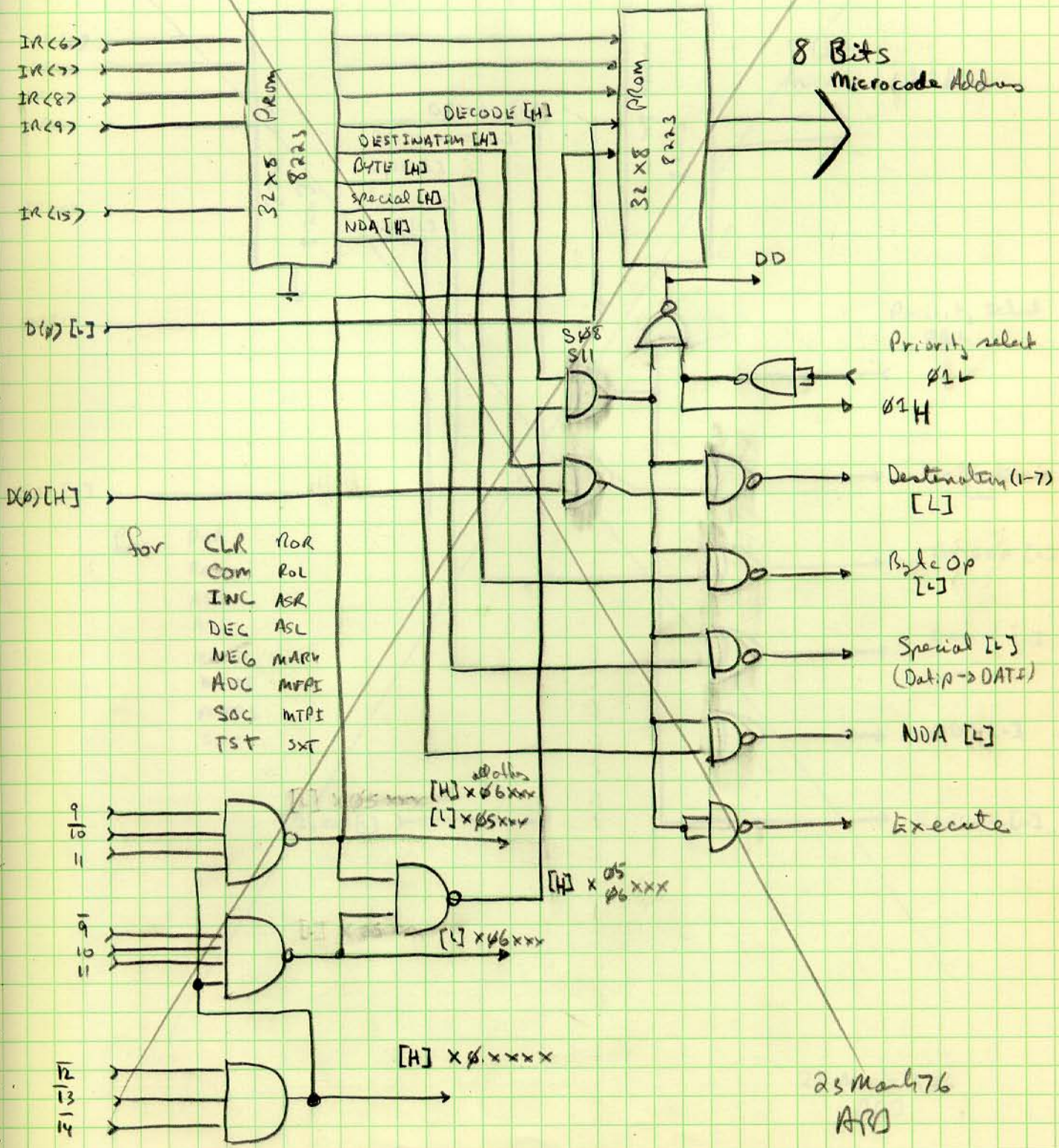
Mode D(Ø)	Destination	Byte op	Special	NDA
CLR(B)	1	IR<15>	1	1
COM(B)	1		1	1
INC(B)	1		1	1
DEC(B)	1		1	1
NEG(B)	1		1	1
ADC(B)	1		1	1
SBC(B)	1		1	1
TST(B)	1		1	1
ROR(B)	1		1	1
ROL(B)	1		1	1
ASR(B)	1		1	1
ASL(B)	1	IR<15>	1	1
MARK	1	1	1	1
MFPI	1	1	1	1
MTPI	1	1	1	1
SXT	1	1	1	1

Mode D(1-7)

CLR(B)	0	IR<15>	1	0
COM(B)	0		1	1
INC(B)	0		1	1
DEC(B)	0		1	1
NEG(B)	0		1	1
ADC(B)	0		1	1
SBC(B)	0		1	1
TST(B)	0		0	1
ROR(B)	0		1	1
ROL(B)	0		1	1
ASR(B)	0		1	1
ASL(B)	0	IR<15>	1	1
MARK	1	1	1	1
MFPI	0	1	1	0
MTPI	0	1	1	0
SXT	0	1	1	1

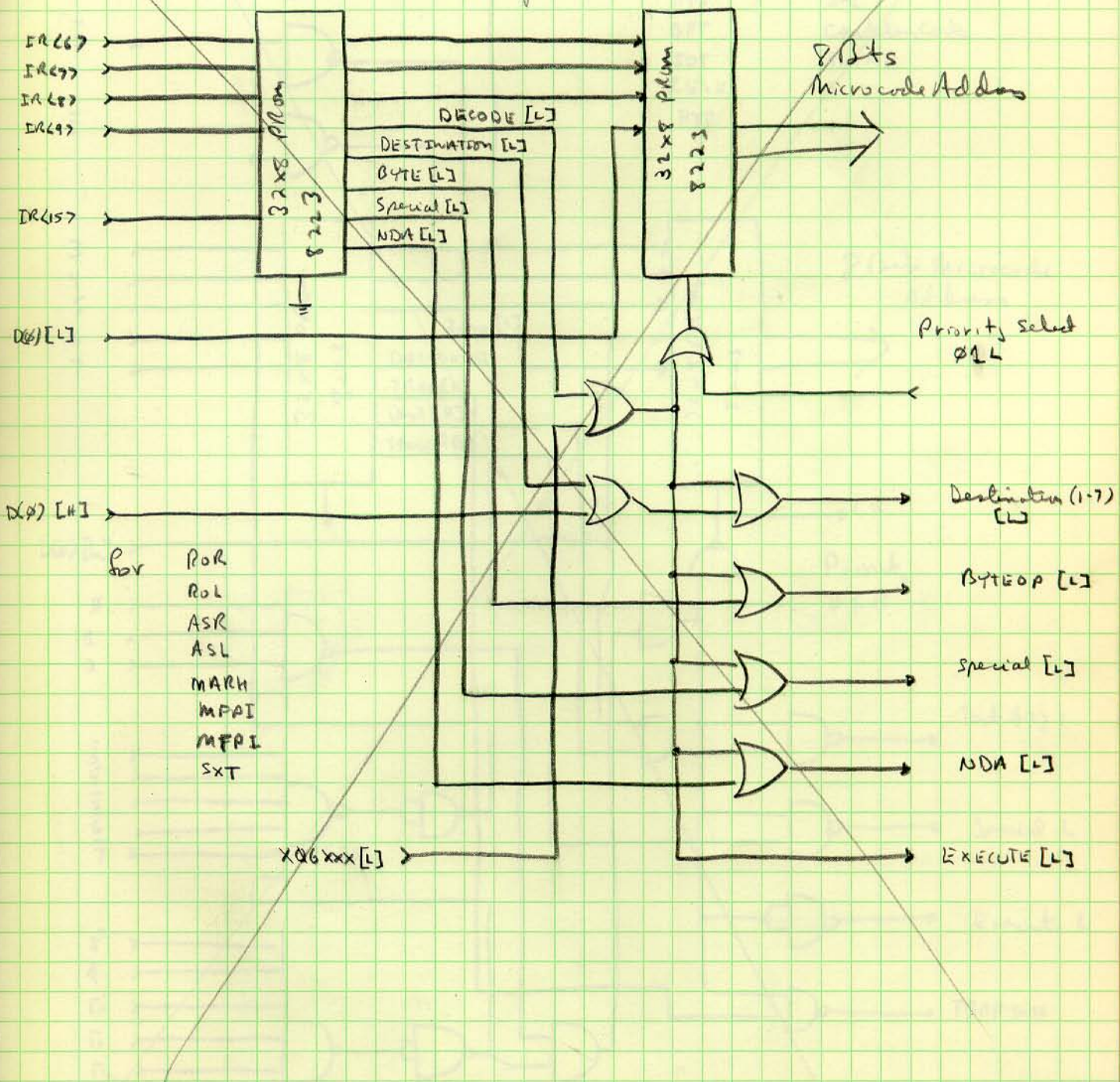
1064xx } Program as No decode
1067xx

Single Operand Decoder # 1



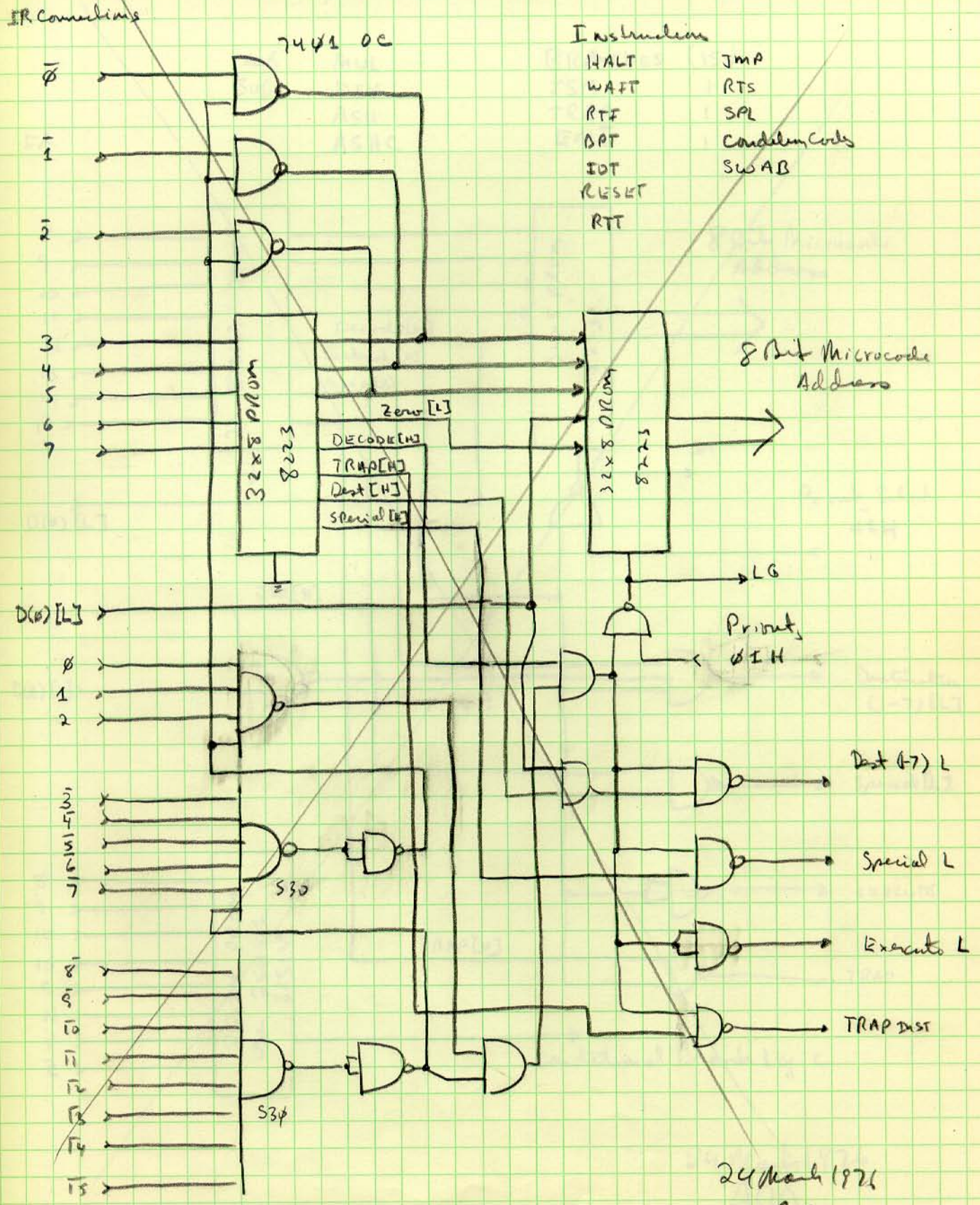
Single Operand Decoder #2

This Decoder not necessary when the previous is used with the modification



23 March 1976
AHD

Low Group Instruction Decoder



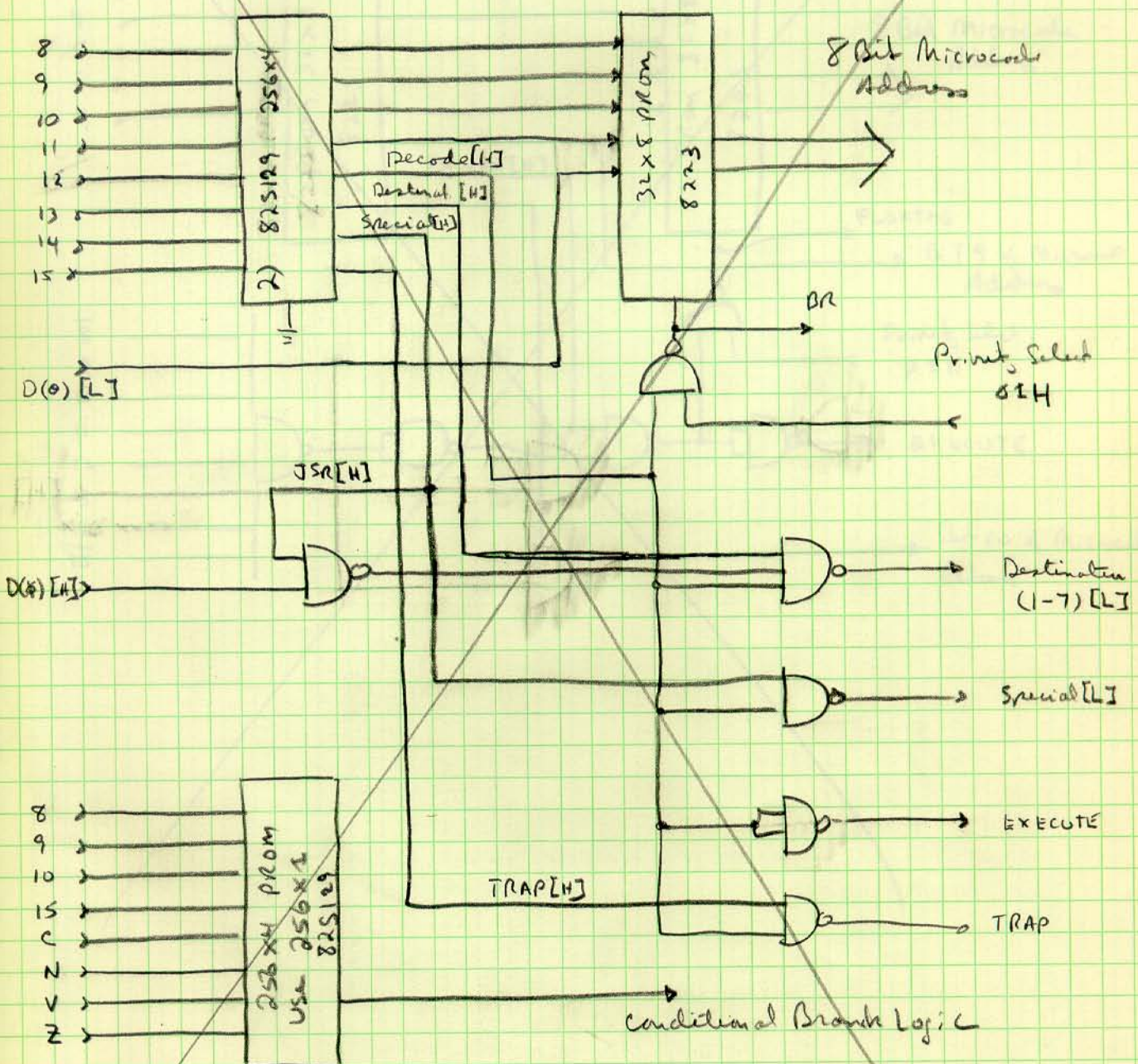
24 March 1976

ARS

Branch, TRAP, and JSR Decoder

XOR	MUL	
SOB	DIV	
	ASH	
	ASHC	

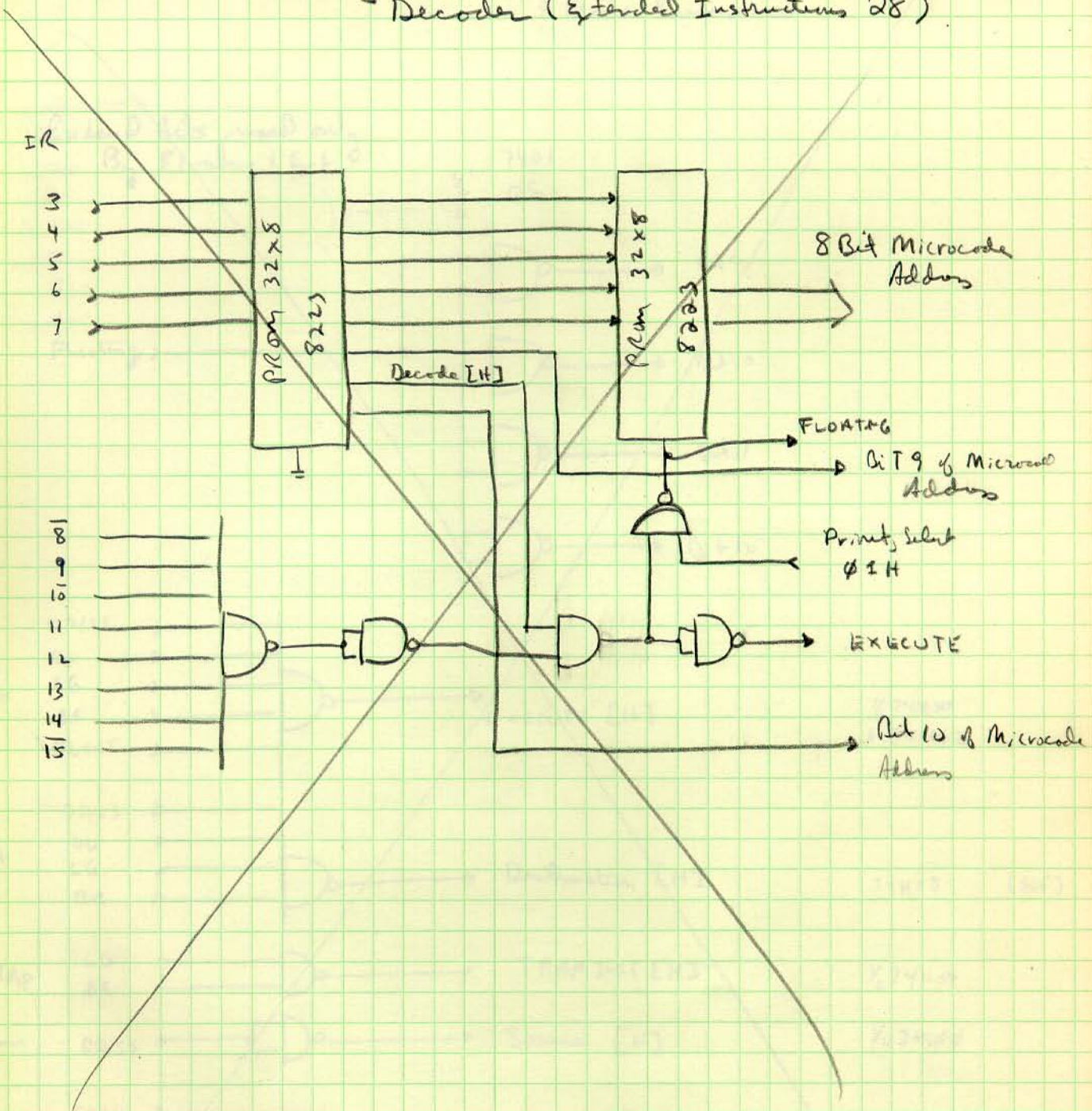
BRANCHES	(15)
JSR	1
TRAP	1
EMT	1



24 March 1976

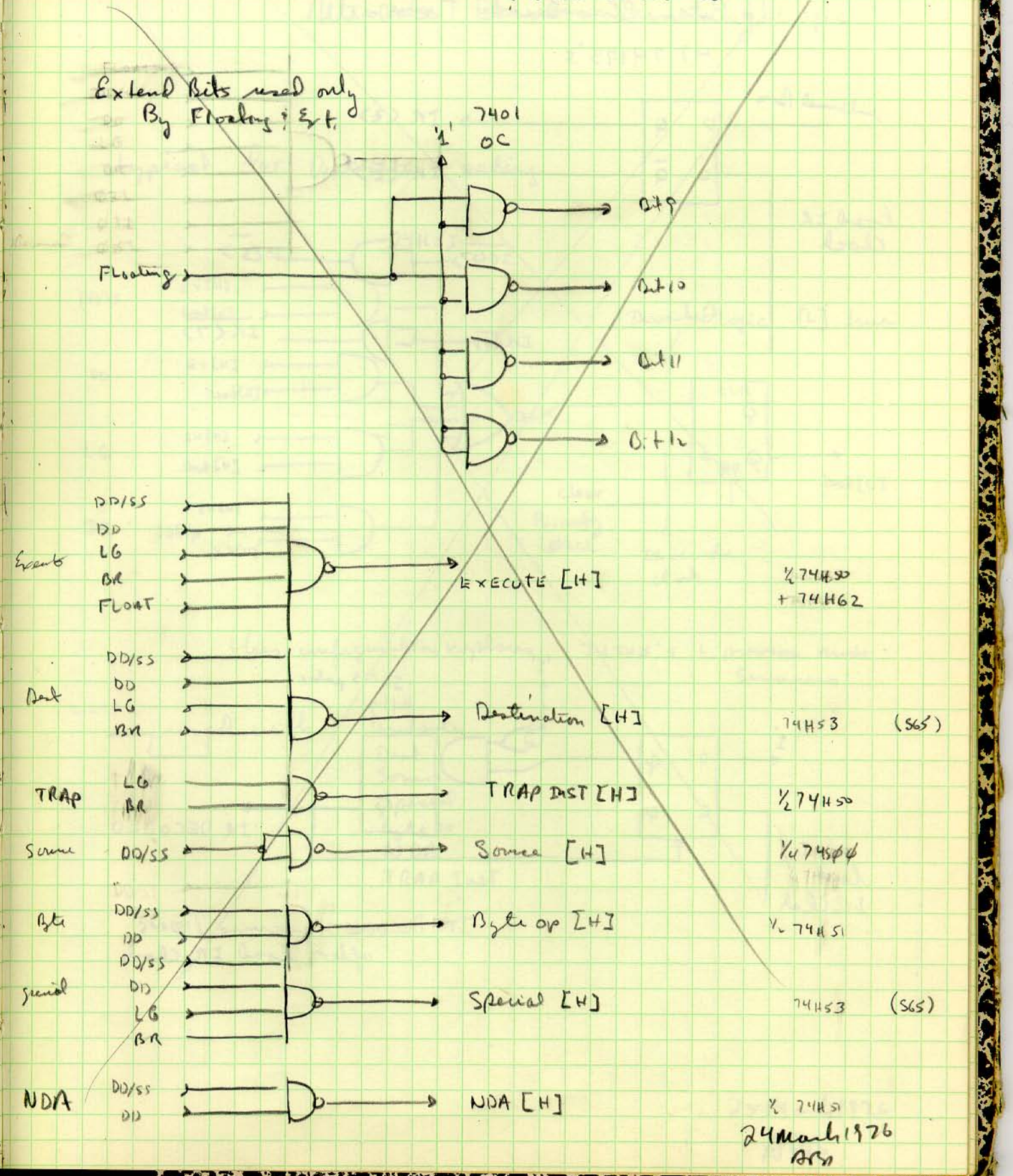
ARD

Floating Point and WCS (Writable Control Store) Decoder (Extended Instructions 28)



24 March 1976
ARB

Bit extension for 12 Bit Microcode Address & Mode Summers

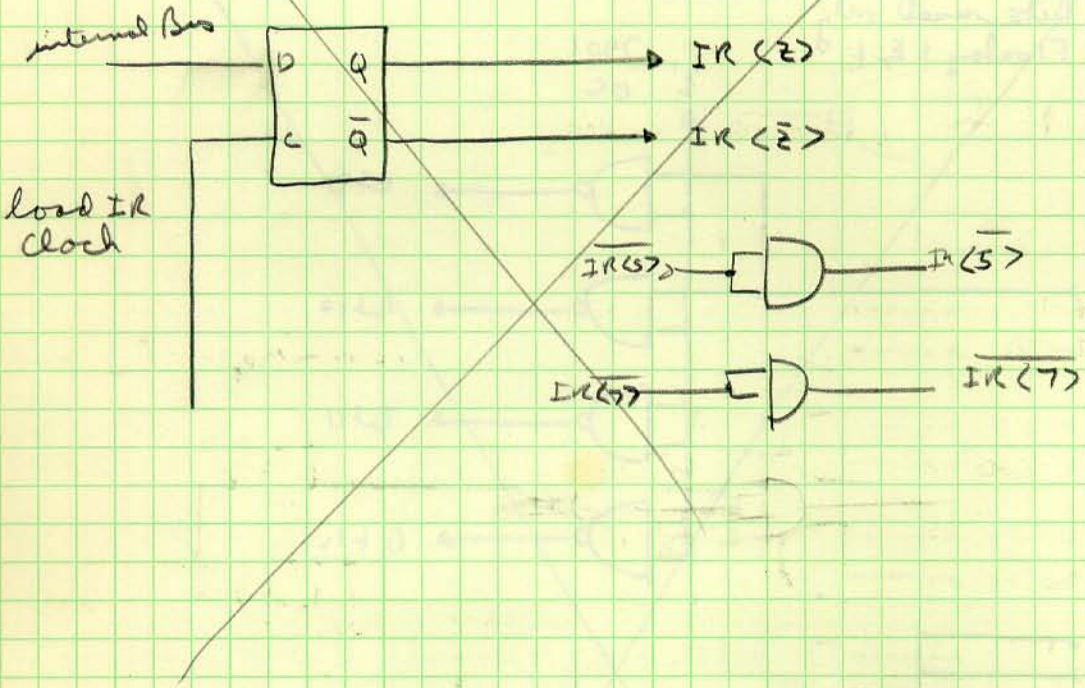


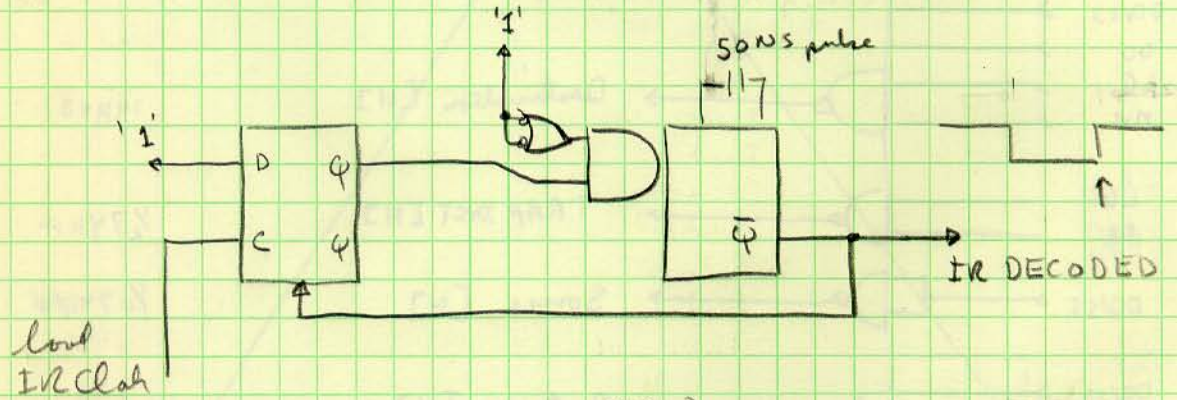
24 March 1976
BSN

IR Register

16 Bits Plus Decode Timer

4) 74175's



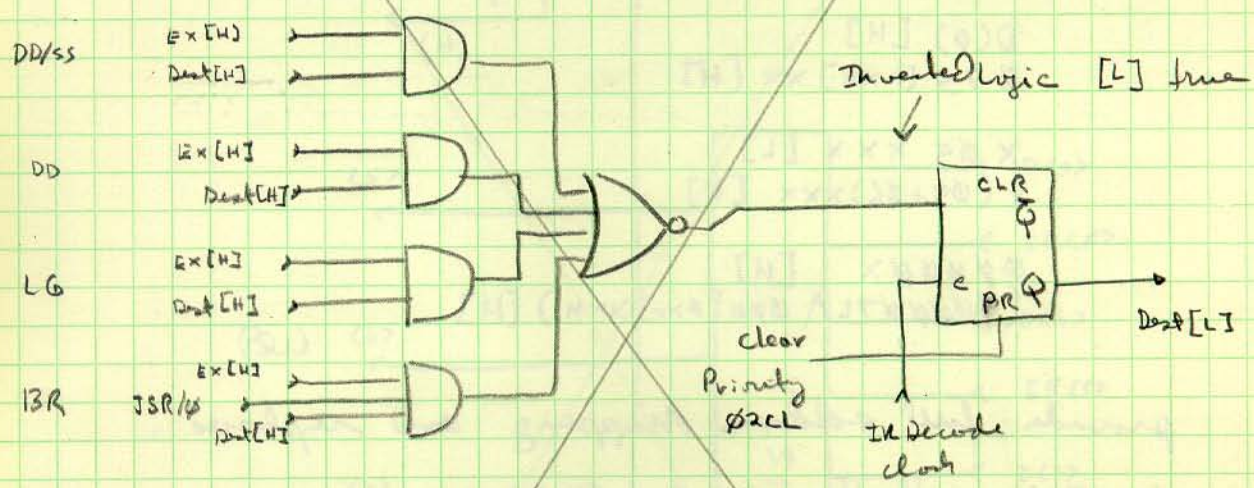


IR DECODE occurs 100NS after load IR clock

Alternate Decoder Summing Logic

typical for Destination coding

74H53



this configuration replaces 74S$\emptyset\emptyset$'s + previous mode Summers

used with execute
Dest
Source
Special
Byte op
NOA
TRAP INST

25 March 1976
ARB

Replace Address selection Logic with the 82S100 (Tri-state) FPLA programmed for the following states

D(ø) [L]
S(ø) [L]

D(ø) [H]
ø 7 5 [ı xx] xx [H]

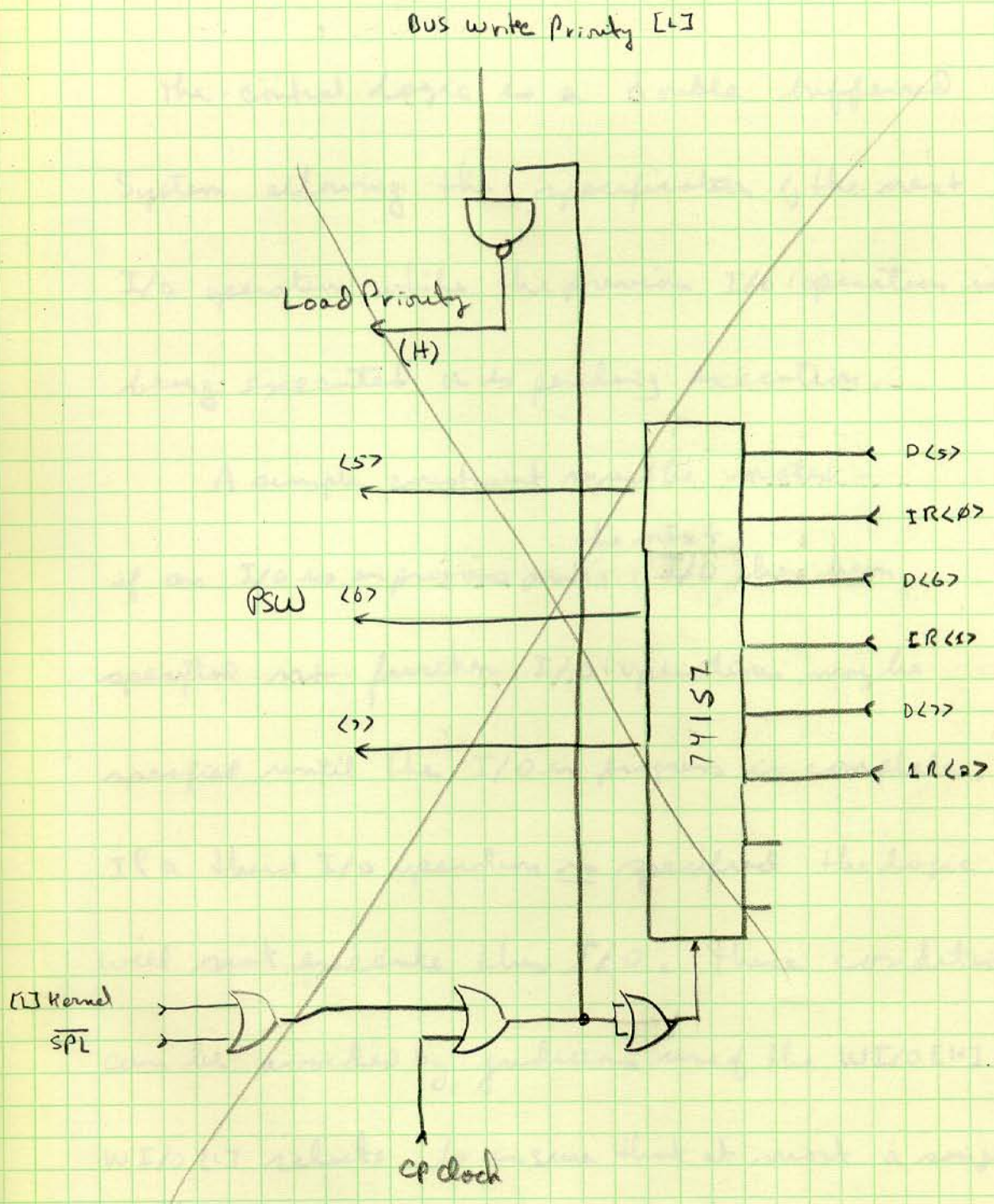
x ø5 xxx [L]
x (ø5+ø6) xxx [H]

ø ø ø ø ø x [H]
(ø ø ø ø ø 7L ∧ ø ø ø [ø xx] xx H) [H]

provides full address Mapping and replaces 10 IC's with the single 82S100

[L]

SPL instruction Logic (optional with the 11/45)
(modifies previous logic)



26 March (P) 6

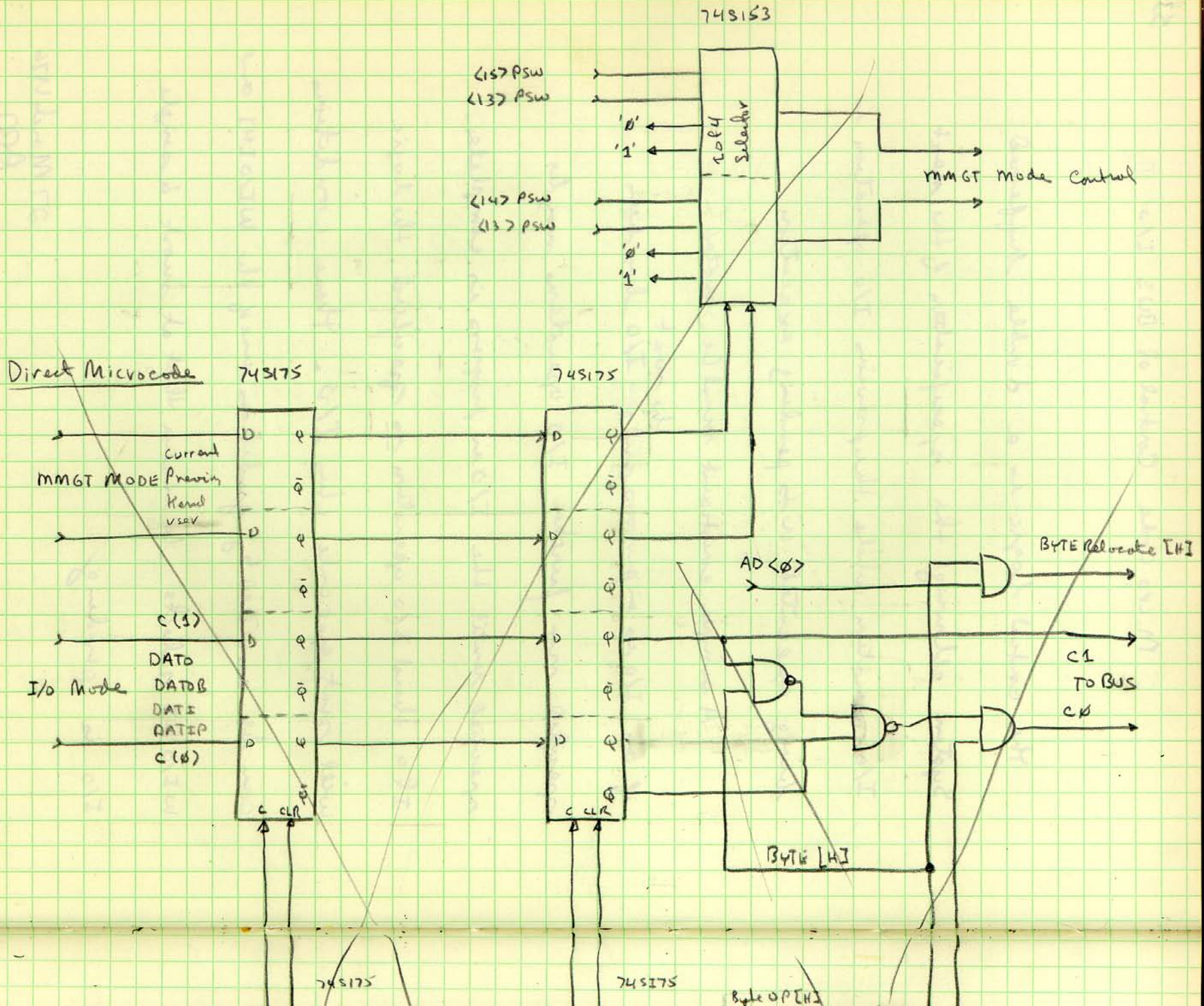
ARB

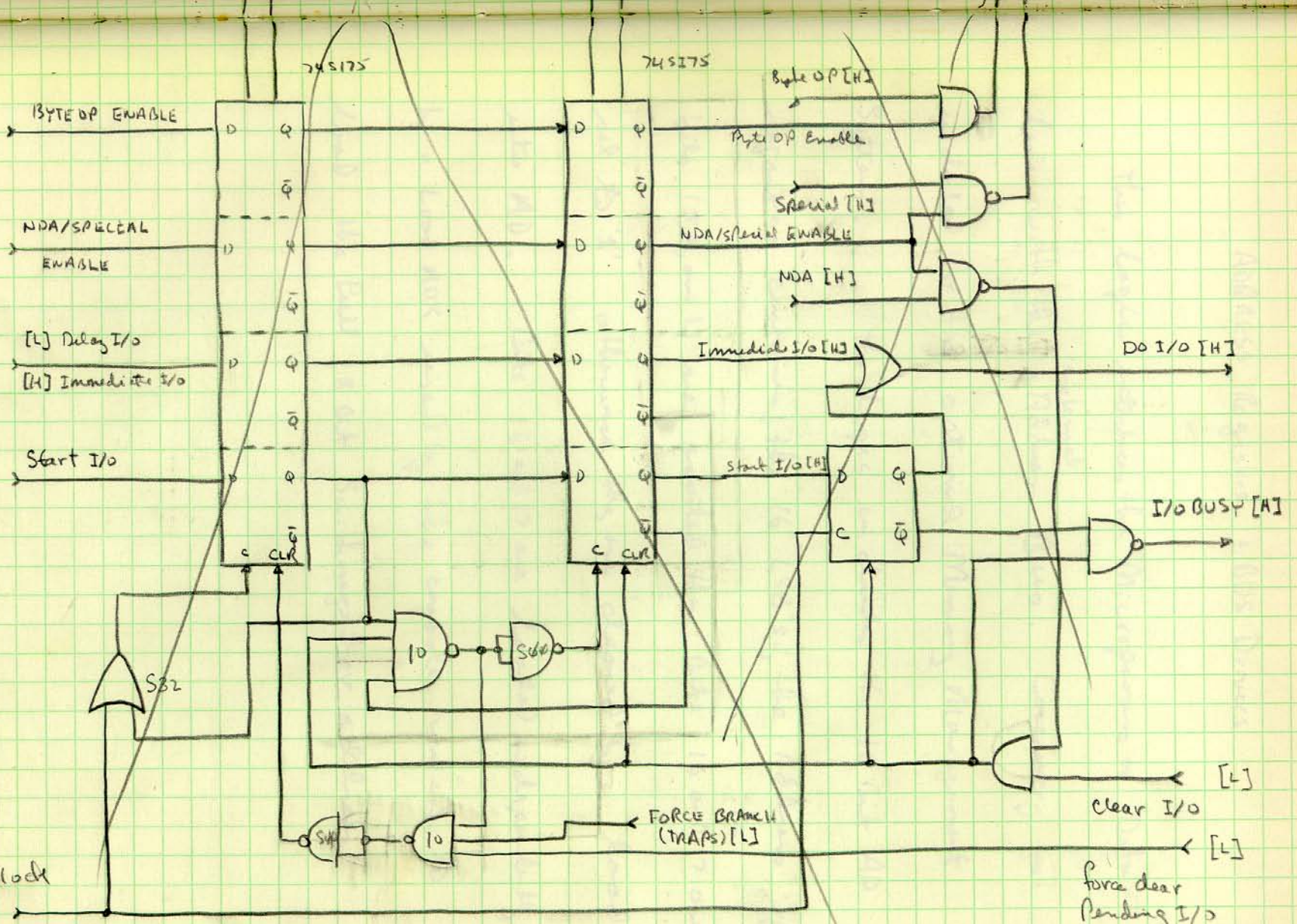
Micro Code Control of BUS I/O

The control logic is a double buffered system allowing the specification of the next I/O operation while the previous I/O operation is being executed or is pending execution.

A simple constraint must be noted - if an I/O is in progress and the next I/O has been specified no further I/O operations may be specified until the I/O in progress is complete. If a third I/O operation is specified the logic will not execute this I/O. These conditions can be avoided by judicious use of the WI/O[4] and WI/O[1] selects to insure that at most a single I/O is pending

27 March 1976
ARD


74S153
<15> PSW
<13> PSW
'0'
'1'
1 of 4 Selector
<14> PSW
<13> PSW
'0'
'1'
MMGT Mode Control
Direct Microcode
74S175
74S175
MMGT MODE
Current
Previous
Kernel
User
C(1)
I/O Mode
DATO
DATOB
DATI
DATIP
C(0)
D
Q
C CLR
AD<0>
BYTE Relocate [HI
C1
TO BUS
C0
BYTE [HI]
74S175
74S175
Byte OP[HI

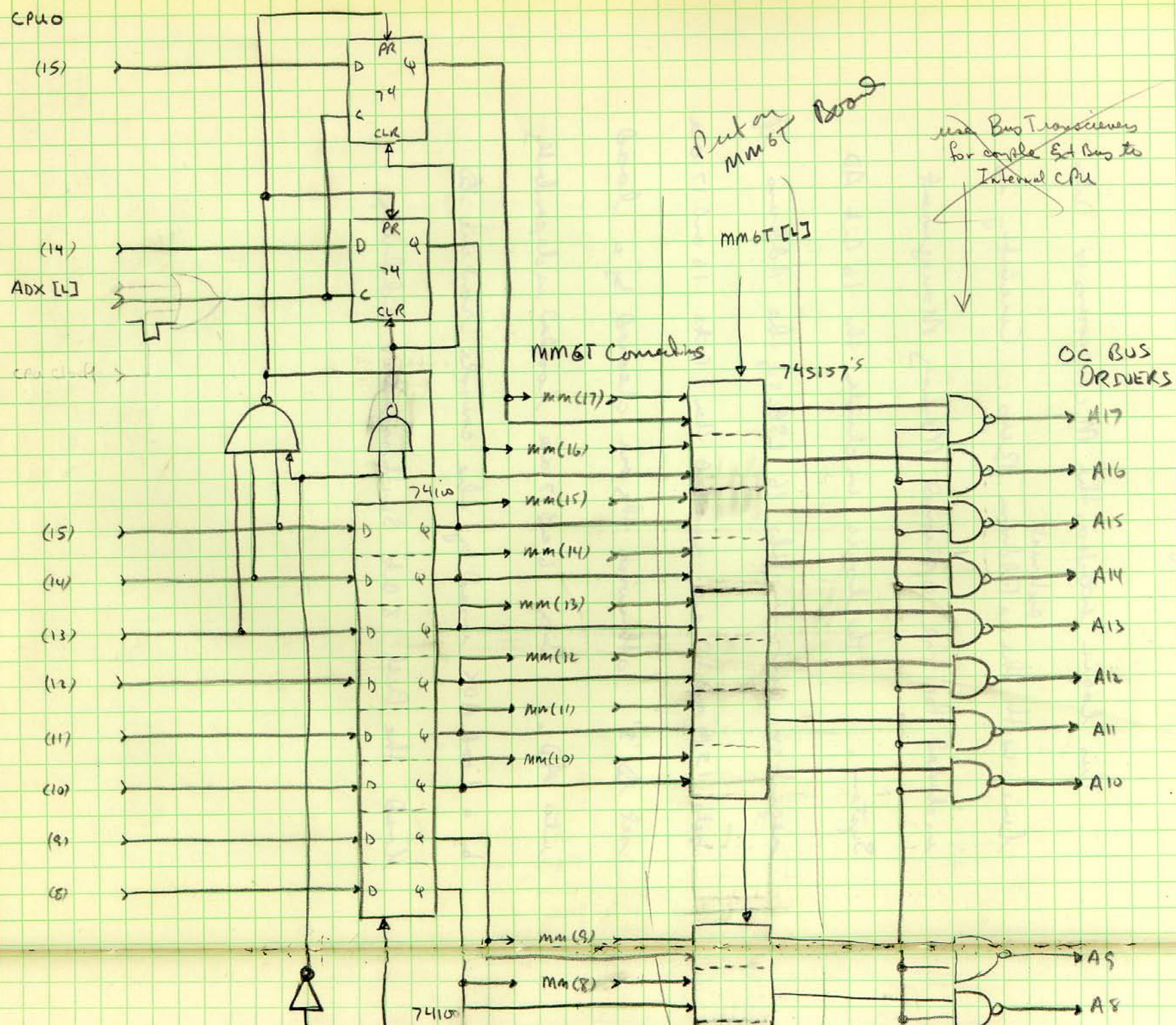


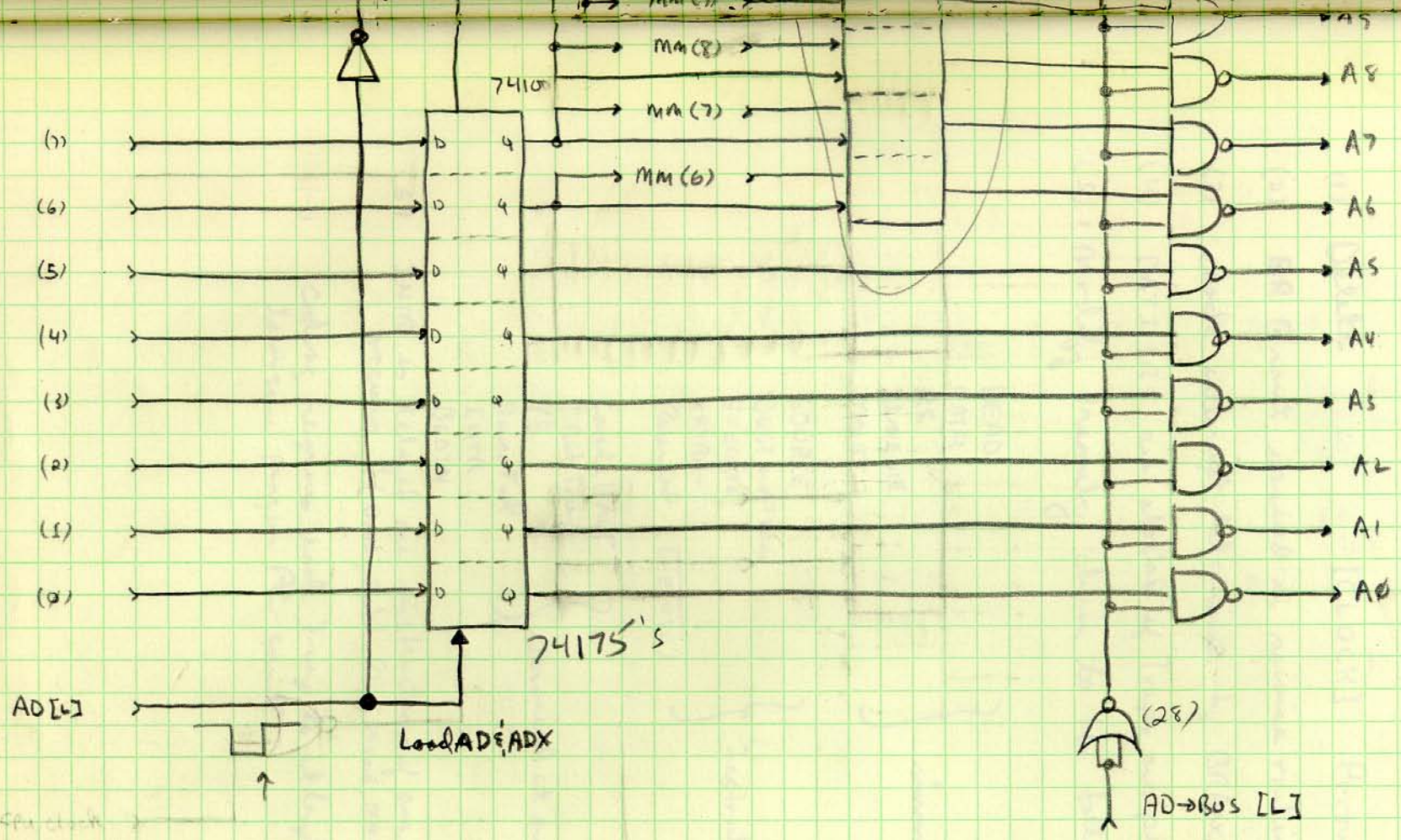
27 March 1976

54

ADDRESS Registers & BUS Drivers

This logic interfaces the Microprocessor Data line with the external Address Bus. Circuitry is included for the optional Memory Management System. The logic includes the 16 Bit AD register containing the 16 LSB's of the Address, if bits 15, 14, and 13 are asserted then Bits 16 and 17 are set to '1' otherwise they are cleared by a load into AD. Bits 16 and 17 are loaded independently by a Load ADX used by the console routine to load the full 18 bit Switchregister address


CPU0
(15)
(14)
ADX [L]
PR
D
74
C
CLR
Q
Put on MM6T Board
MM6T [L]
MM6T Connections
74S157's
OC BUS DRIVERS
mm(17)
mm(16)
mm(15)
mm(14)
mm(13)
mm(12)
mm(11)
mm(10)
mm(9)
mm(8)
74LS10
(15)
(14)
(13)
(12)
(11)
(10)
(9)
(8)
A17
A16
A15
A14
A13
A12
A11
A10
A9
A8



7410 must be replaced with edge triggered latches

ADD

27 March 1974

Modification in Interrupt Handling

(1) Delete [CLOCK] Priority from chain

(2) BR Grant issued in service routine

(3) clock interrupt determined by BRANCH on TEST

(4) DATI IR's are aborted if Traps are pending

(5) Modify Priority chain to the following

DEAD
FATAL
RS
TIMEOUT
MMGT
} immediate Branches

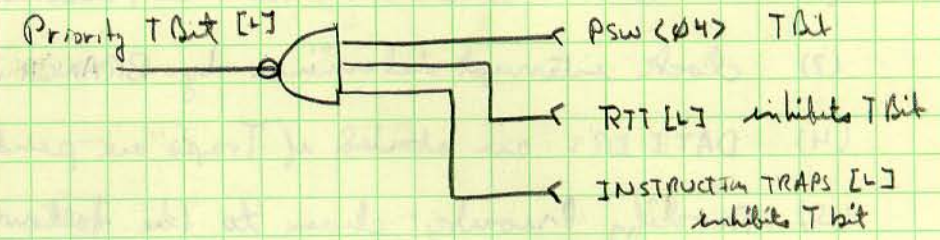
SOURCE
DESTINATION
EXECUTE
EXTRA
Reserved ($\overline{\text{EXECUTE}}$)
} Instruction Execution

Console HALT
T Bit Trap
YS
Power Fail
} Serviced at end of Instructions
INTR
BEGIN

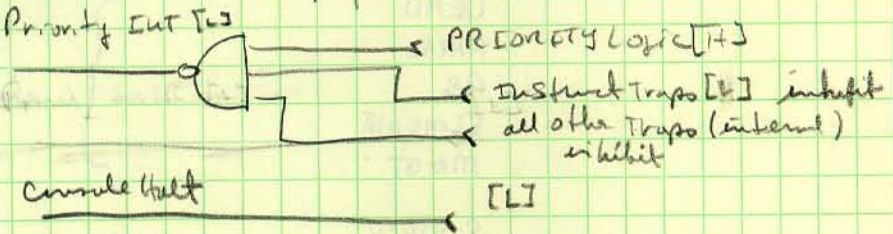
(6) INTR is delayed one instruction if an internal Trap occurs (Dead, Fatal, RS, TIMEOUT, MMGT, T Bit, YS, PF, Console)

(7) Coding requires each Trap to attempt a DATI IR to insure proper PC count

The Trace Trap Logic is simplified as follows (straight DC level is sufficient for Trace)



Interrupt Priority logic is also simplified



11/35 TBit operations

RTI setting T Bit , immediate Trace Trap
RTT setting T Bit , delayed Trace Trap
Instruction Traps , delayed Trace Trap
if an IntR occurs between T set and Trap – complete Routine executed
on RTI or RTT above occurs
if T bit was Set and RTI or RTT is the Traced instruction action depends upon PSW loaded

INTR Routine

* clock → DO ; DATI INT ; BR TO TRAP

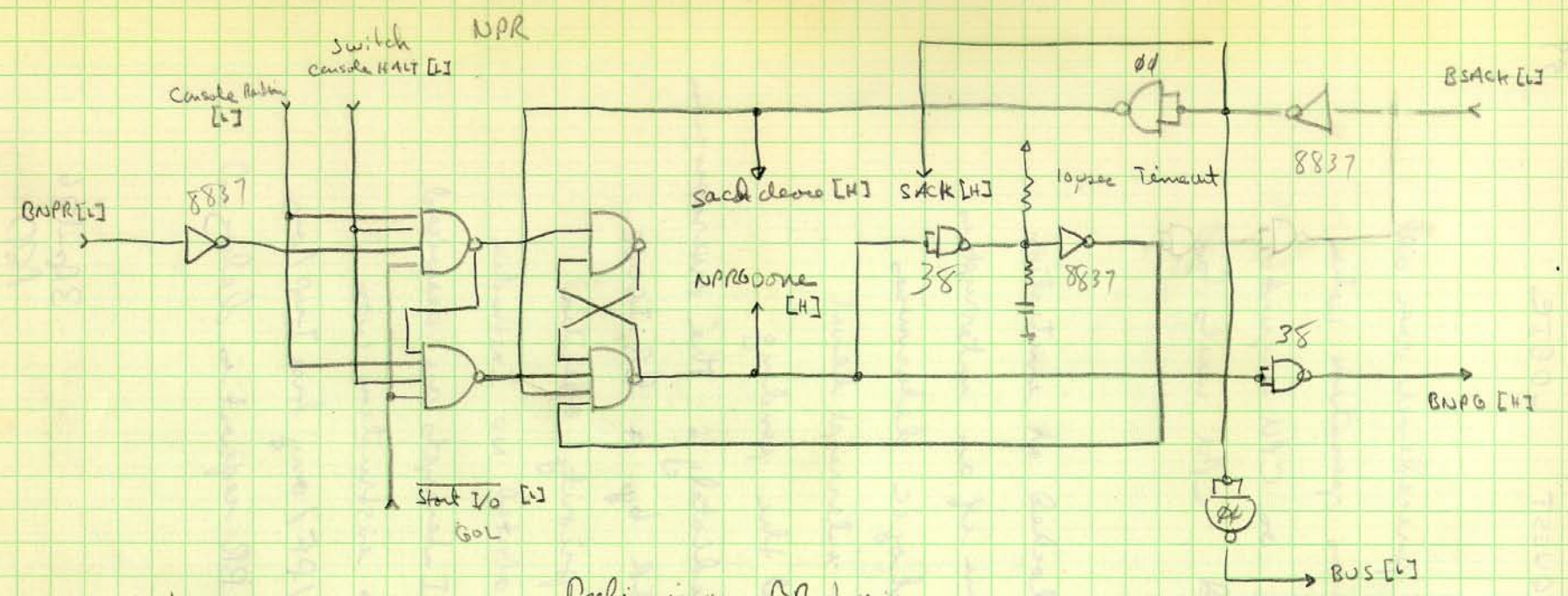
if NO BR's, DATI INT will be aborted with # clock left in DI; if a BR level is enabled then the external Vector will be loaded into DI

NPR & BRANCH REQUEST LOGIC

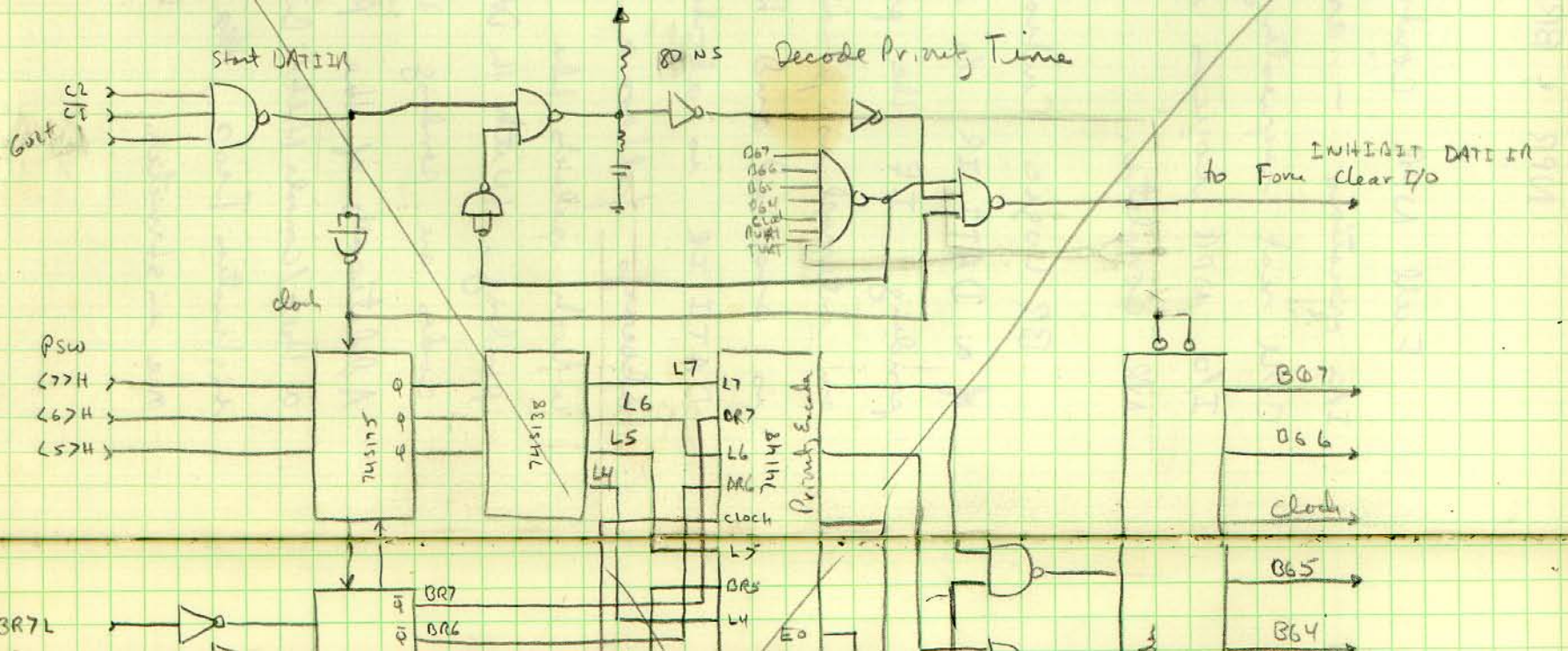
Full NPR control - precedence over all I/O operations - asynchronous operation when CPU not requesting I/O; on CPU request I/O NPR logic is latched, CPU waits for NPR complete.

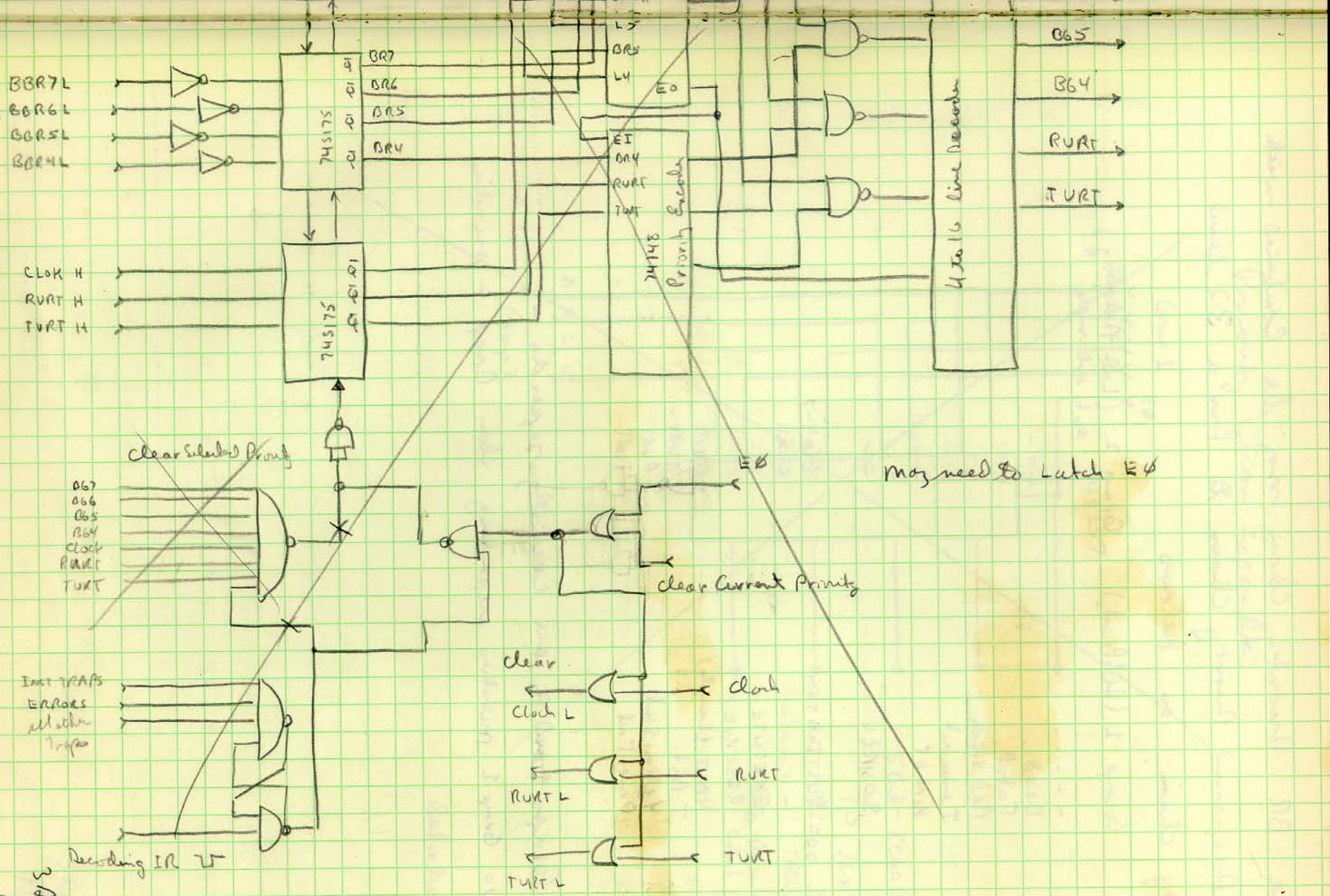
BR Logic Control is clocked at initiation of a DATI IR to determine if an interrupt is pending. If the priority logic determines an interrupt is pending the interrupt level is asserted and held and the pending DATI IR is aborted immediately. The microprogram always follows a DATI IR by a BR ICODE which selects the highest priority operation pending. With the DATI IR aborted NO Instruction codes are pending therfore Interrupts are serviced. Additionally if the previous instruction was a Halt/Console Halt/T Bit/YS/PF/any trace Trap/any instruction Trap, ~~the~~ any BR request is delayed one instruction.

3April76
ABD



Preliminary BR Logic




BBR7L
BBR6L
BBR5L
BBR4L
74S175
BR7
BR6
BR5
BR4
CLOK H
RURT H
TURT H
74S175
L7
BR5
L4
EO
EI
BR4
RURT
TURT
74148
Priority Encoder
4 to 16 line decoder
BG5
BG4
RURT
TURT
clear Silenced Prioty
BG7
BG6
BG5
BG4
Clock
RURT
TURT
EØ
clear Current Prioty
may need to Latch EØ
INST TRAPS
ERRORS
all other
Traps
Decoding IR 7
clear
Clock L
clock
RURT L
RURT
TURT L
TURT
3 April 76
ADD

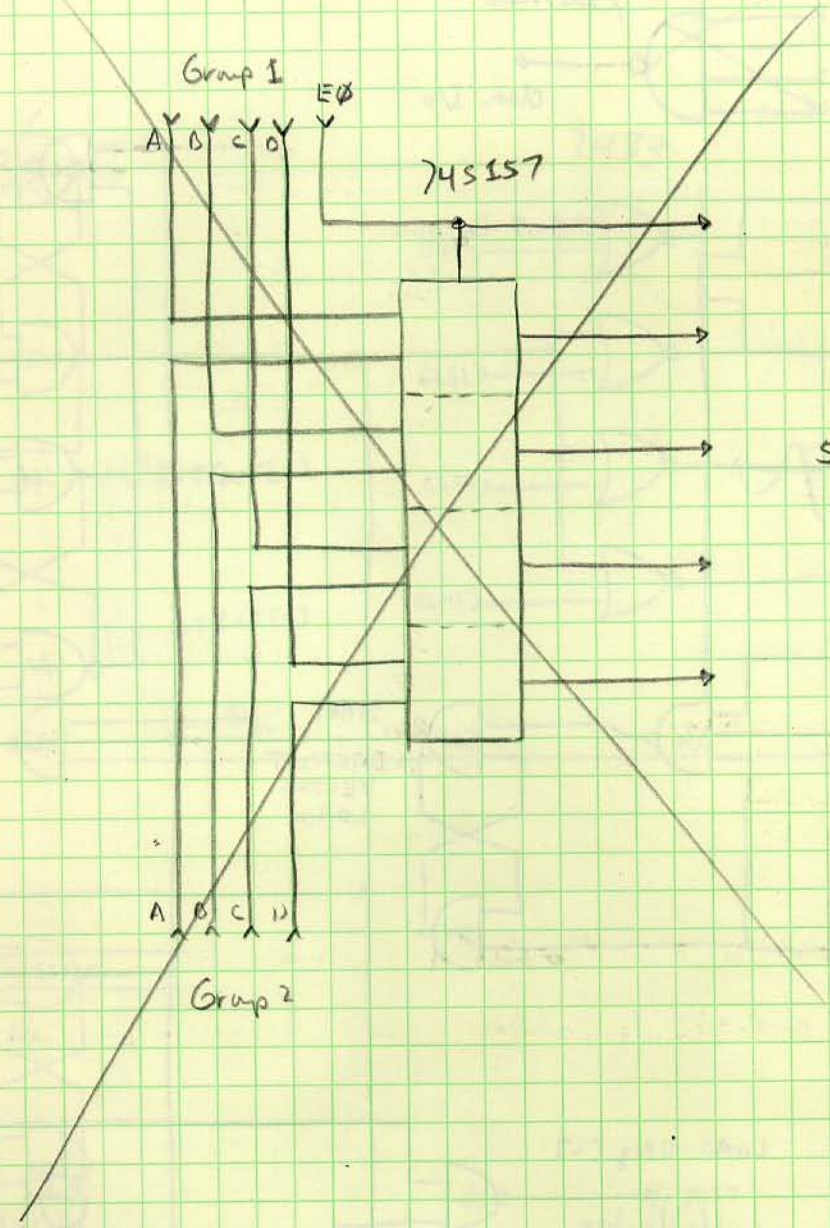
The BR Priority chain may be combined with the Programmed Priority chain to form a 32 level Priority chain as follows

	Grup 1 (High Priority)	Grup 2 (Low Priority) ext. interrupts
1	- - - -	- - - -
2	Dead	- - -
3	Fatal	- -
4	Red Stuck	- -
5	Time out	- -
6	MM 6T	- -
7	- - -	- - -
8	SOURCE	- -
9	DESTINATION	BG4-7
10	- - - -	CLOCK
11	EXECUTE	
12	RESERVED	
13	HALT (Console)	1 RUPT
14	T Bit	1 TURT
15	Yellow Stuck	2 RUPT
16	Power Fail	2 TURT

Grups 1 and 2 are logically seperate -

if No Grup 1 priorities are pending than Grup 2 priorities are selected

Priority Chain Expander



5 Bit ICODE

TO 32x8

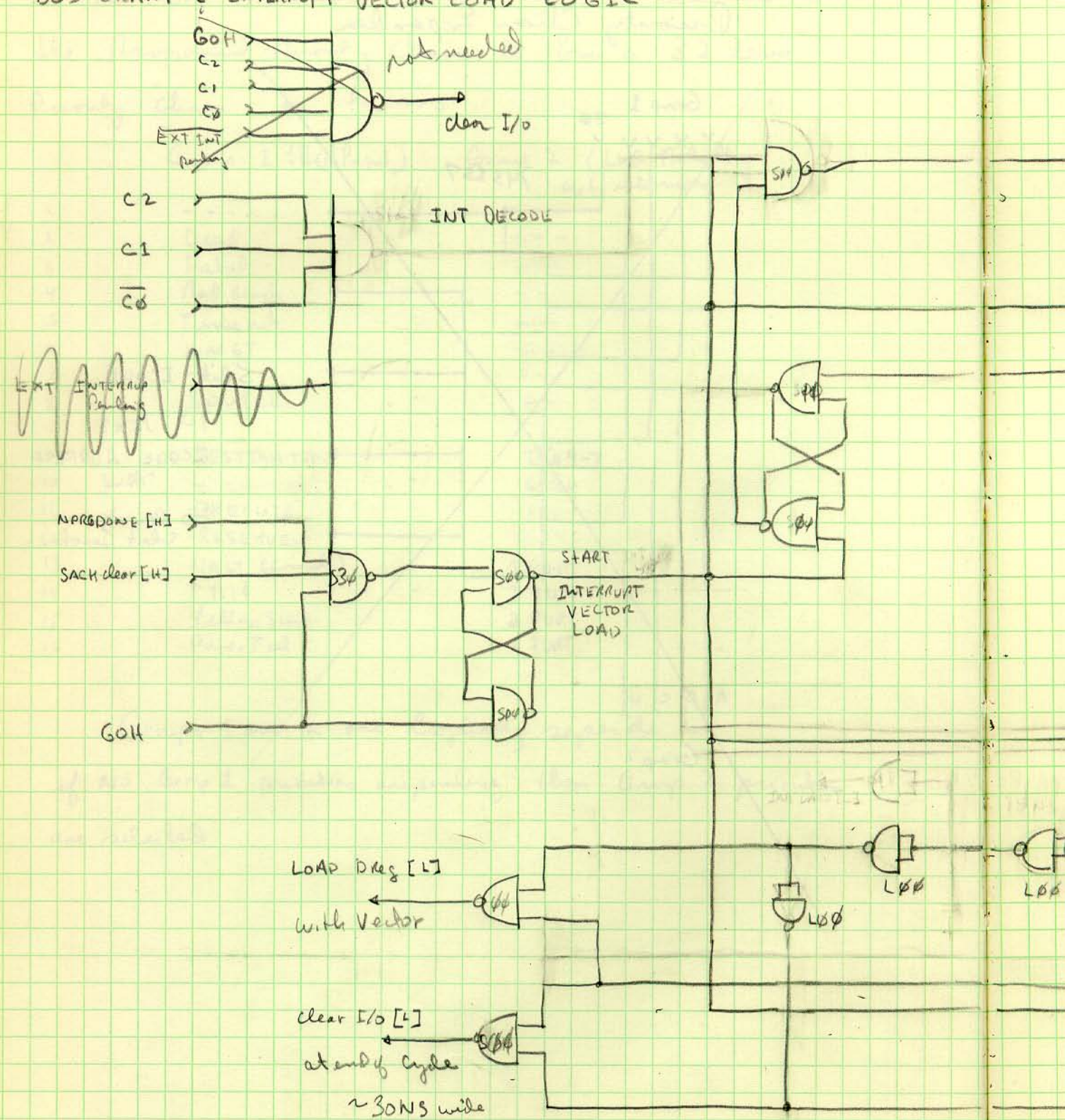
ICODE ADDRESS TABLE

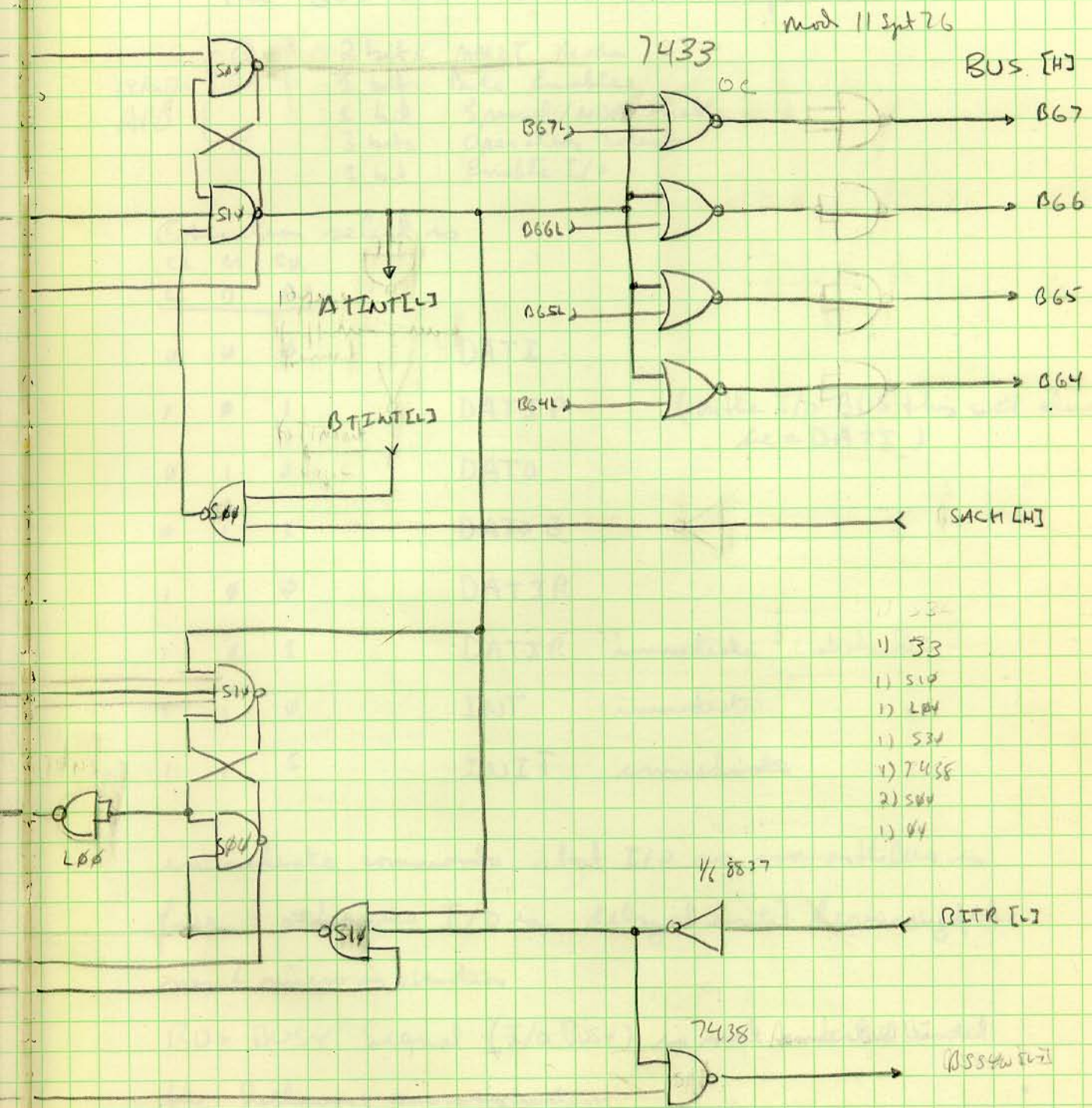
AND Priority Select Decoder

3 April 76

ARD

BUS GRANT & INTERRUPT VECTOR LOAD LOGIC





1) 33
1) S10
1) L04
1) S30
4) 7438
2) S00
1) 04

4 April 76
ARD

Alternate BUS Control Decode

The BUS code is on 8 10 Bit segment

28 Aug 76 ARB

③ 2 bits MMGT Mode
1 bit Byte Enable
1 bit Special / NDA enable

④ 3 bits Operation Select
1 bit Enable I/O

Operation select is

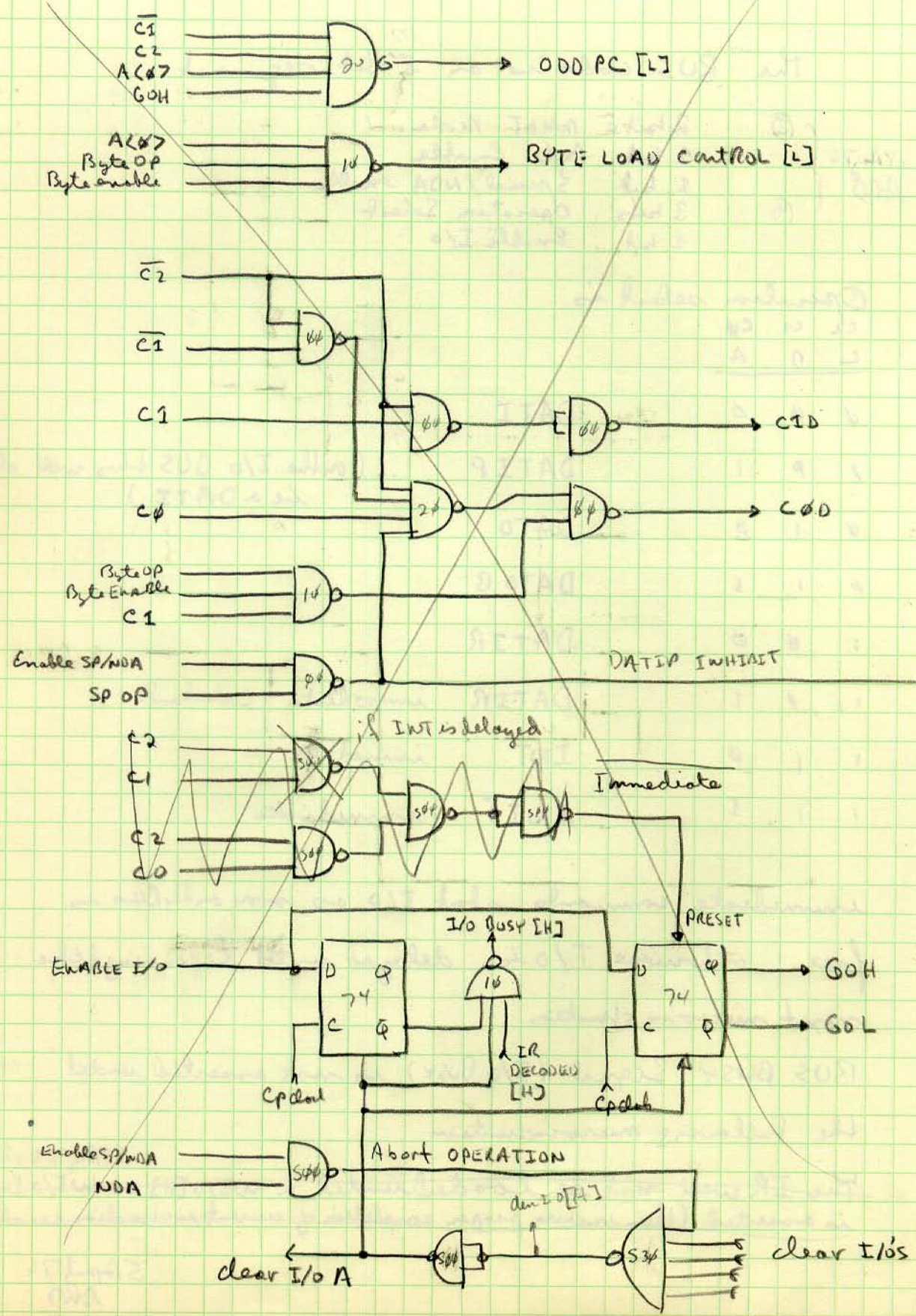
C_2	C_1	C_0	
C	B	A	
0	0	0	DATI
0	0	1	DATIP (on the I/O BUS this will always be a DATI)
0	1	0	DATO
0	1	1	DATOB
1	0	0	DATIR
1	0	1	DATIR immediate (lookahead)
1	1	0	INT immediate
1	1	1	INIT immediate

immediate commands start I/O as soon as the Bus is free otherwise I/O is delayed until beginning of the next micro instruction

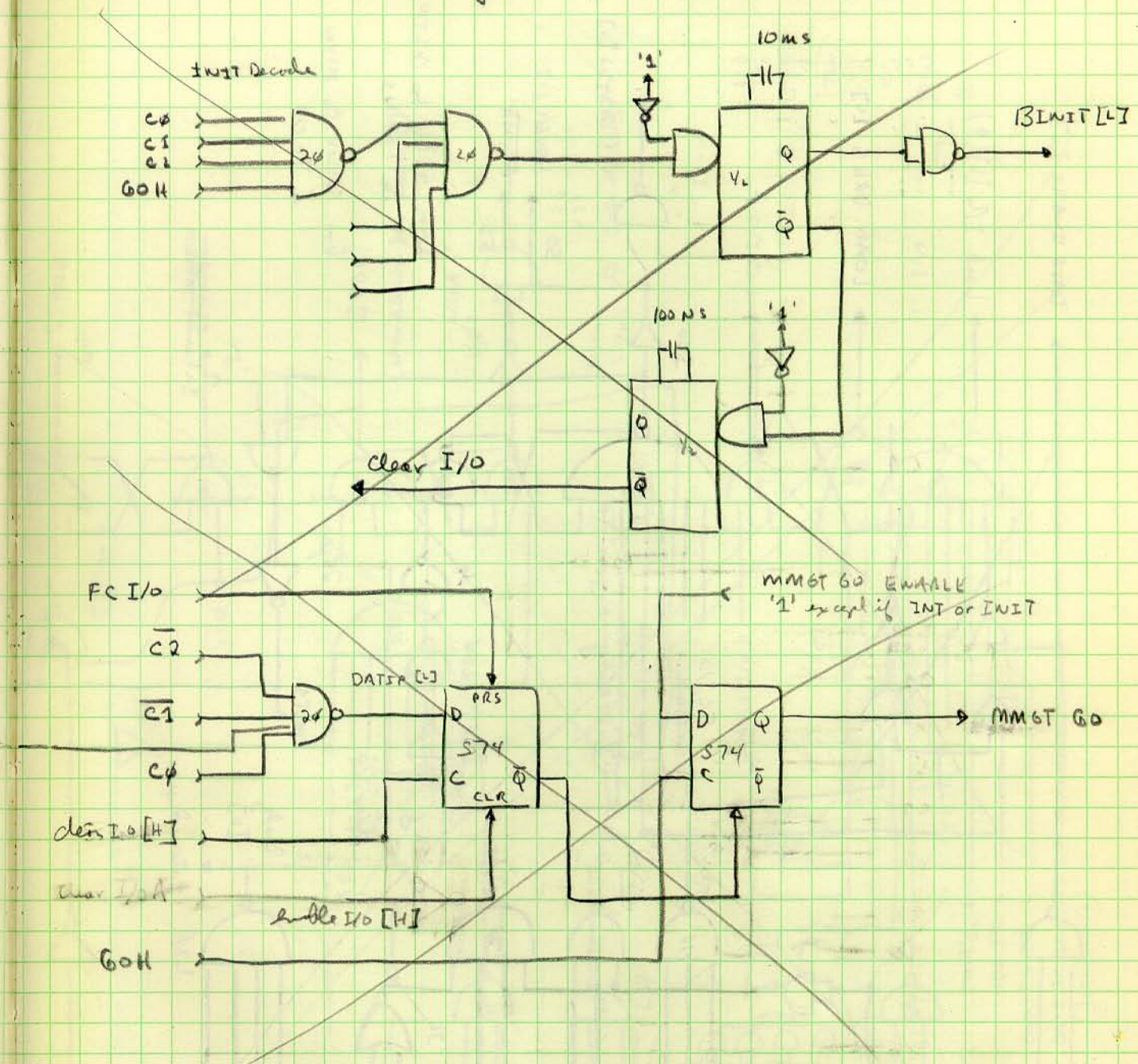
BUS BUSY Signal (I/O BUSY) is not asserted until the following microinstruction

28 Aug 76 ARB
~~The IR will not be loaded until a WI/O [H] or WI/O [L] is asserted (this insures proper completion of current instruction decoding)~~

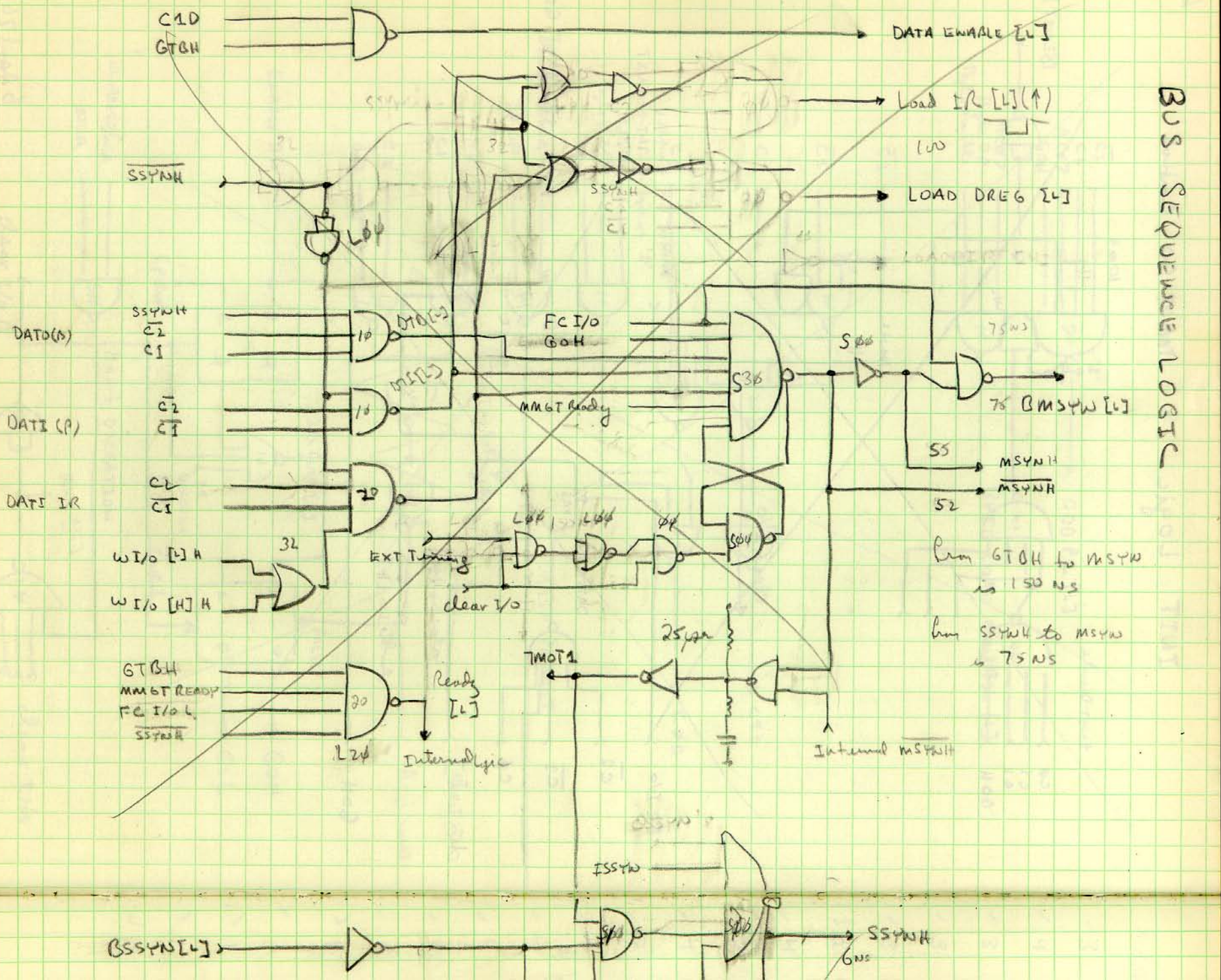
5 April 76
ARB

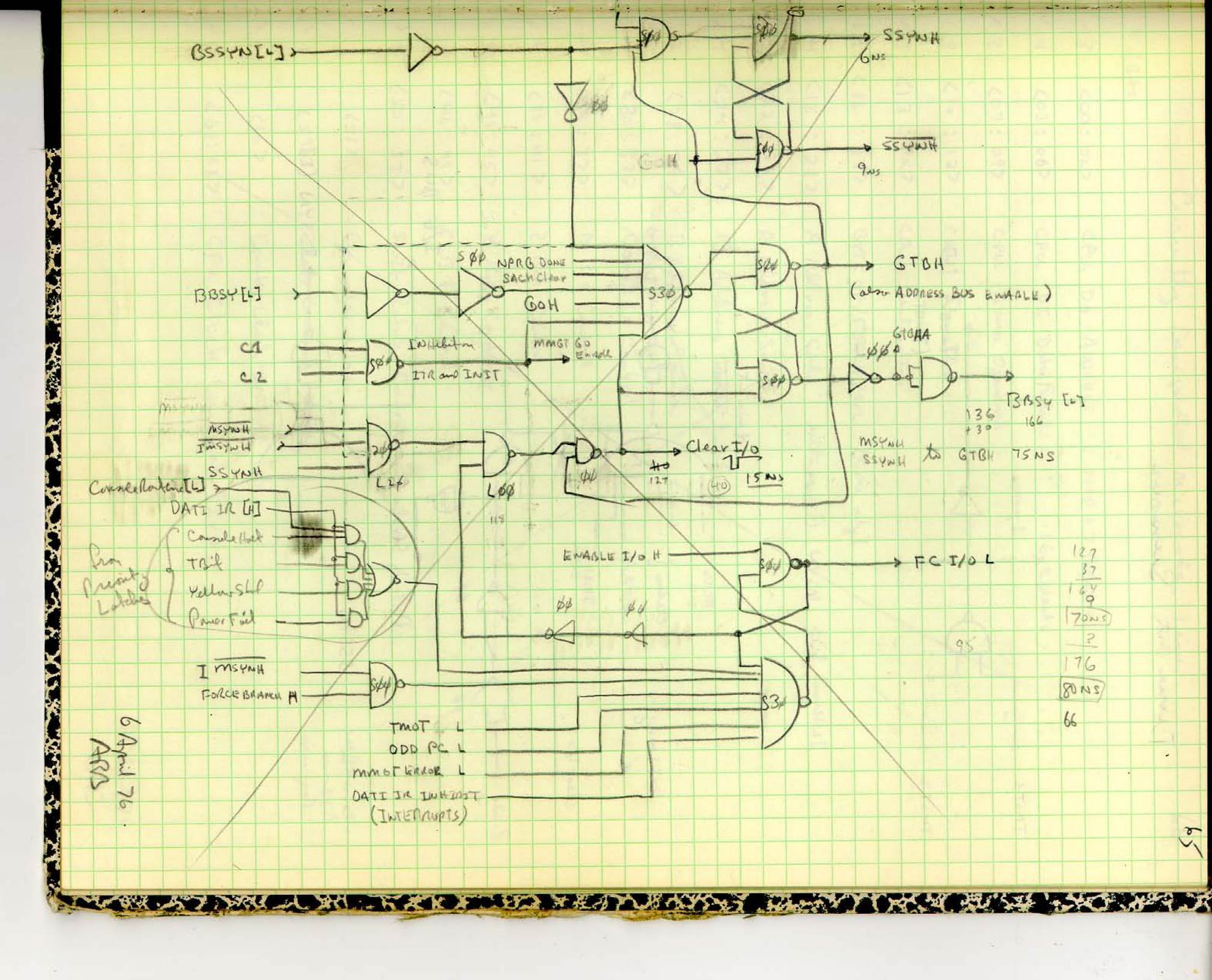

C1
C2
A(07)
GOH
ODD PC [L]
A(07)
Byte OP
Byte enable
BYTE LOAD Control [L]
C2
C1
C1
C0
C1D
C0D
Byte OP
Byte Enable
C1
Enable SP/NDA
SP OP
DATIP INHIBIT
if INT is delayed
C2
C1
C2
C0
Immediate
I/O Busy [H]
PRESET
ENABLE I/O
D
Q
74
C
Q
GOH
GOL
IR DECODED [H]
Cpclock
Cpclock
Enable SP/NDA
NDA
Abort OPERATION
clear I/O A
clear I/Os

INIT Logic & MM6T GO Control

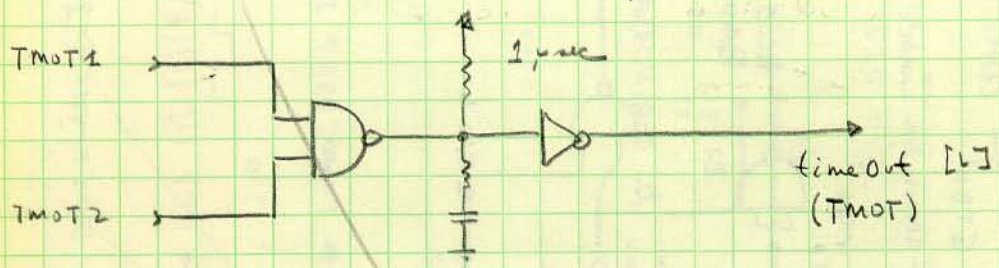


6 April 76
ARB

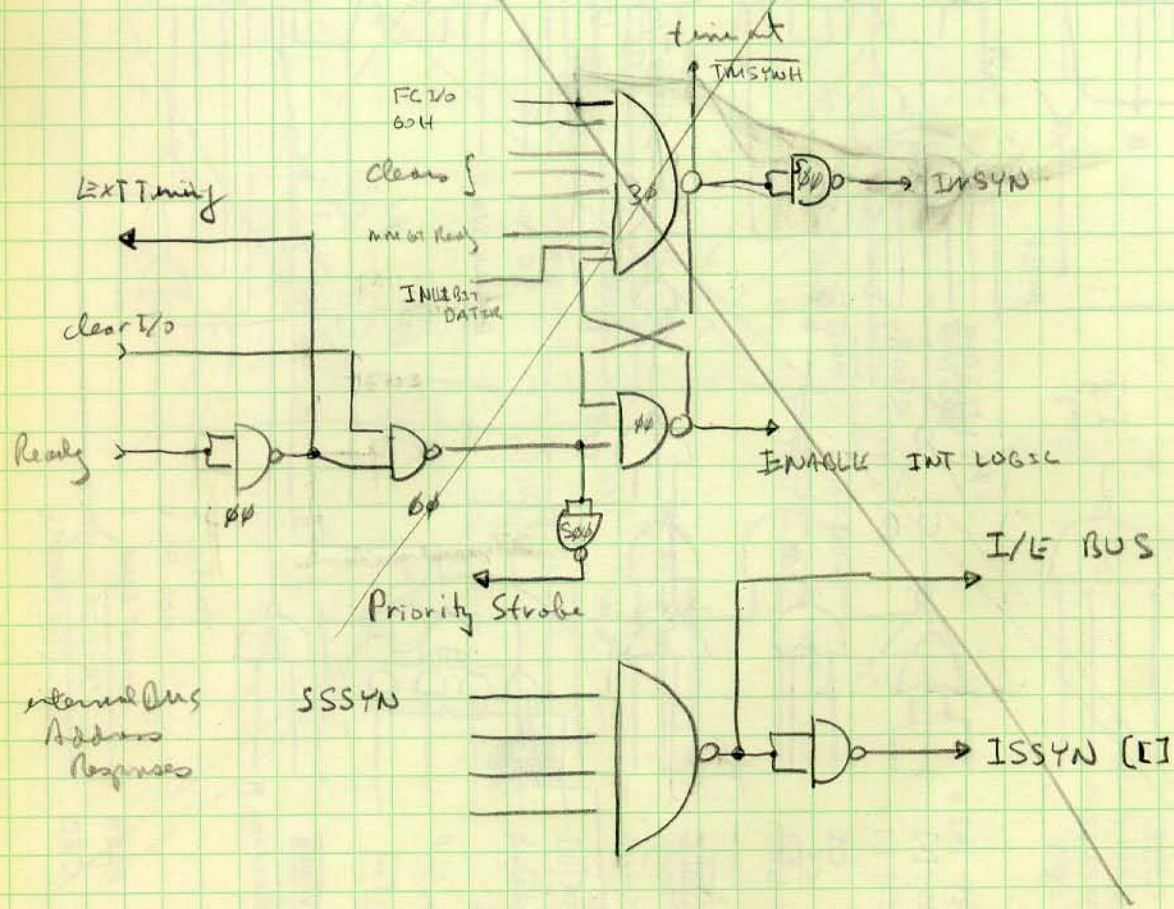

BUS SEQUENCE LOGIC
C1D
GTBH
DATA ENABLE [L]
Load IR [L] (↑)
LOAD DREG [L]
SSYNH
L0φ
DATO(N)
DATI (P)
DATI IR
FC I/O
GOH
MMGT Ready
S36
S φφ
75 BMSYN [L]
55
MSYN H
MSYNH
52
EXT Timing
clear I/O
W I/O [L] H
W I/O [H] H
GTBH
MMGT READY
FC I/O L
Ready [L]
L2φ
Interndlgic
7MOT1
25μs
Internal MSYNH
ISSYN
BSSYN [L]
SSYNH
6NS
from GTOH to MSYN is 150 NS
from SSYNH to MSYN is 75 NS


BSSYN[L]
SSYNH
6NS
SSYNH
9NS
GOH
SØØ NPRG DONE
SACH Clear
GOH
S30
BBSY[L]
C1
C2
Inhibit on
ITR and INIT
MMGT GO Enable
GTBH
(also ADDRESS BUS ENABLE)
GTBAA
BBSY [L]
136
+30
166
MSYNH
SSYNH
to GTBH 75NS
MSYNH
IMSYNH
SSYNH
L2Ø
LØØ
Clear I/O
15NS
ConsoleHalted[L]
DATI IR [H]
ConsoleHalt
TRit
YellowStkf
PowerFail
From Present Latches
ENABLE I/O H
FC I/O L
I MSYNH
FORCE BRANCH H
TMOT L
ODD PC L
MMGT ERROR L
DATI IR INHIBIT
(INTERRUPTS)
127
37
164
9
70NS
2
176
80NS
66
95
April 76

Time out Summer



Internal BUS MSYN Generation



Currently Designated Microcode Lines

# of Bits		
3	<00:02>	CPU DO ADDRESS — NOP + 7 Registers
4	<03:06>	CPU DI Data Select — 16 selects
3	<07:09>	CPU Source Operand
3	<10:12>	CPU Function
3	<13:15>	CPU Destination
2	<16:17>	BYTE OP/FORCE/SPL [CPU only]
4	<18:21>	A Address Code
2	<22:23>	A Address Source
4	<24:27>	B Address Code
2	<28:29>	B Address Source
4	<30:33>	Carry IN Control — 16 states
4	<34:37>	N Bit Control
4	<38:41>	Z Bit Control
4	<42:45>	V Bit Control
4	<46:49>	C Bit Control
8	<50:57>	Shift Rotate Control
1	<58>	Console Line
1	<59>	Clear Current Priority
1	<60>	Counter Clock Control
2	<61:62>	CPU Clock Control (I/O)

6 April 76
ARB

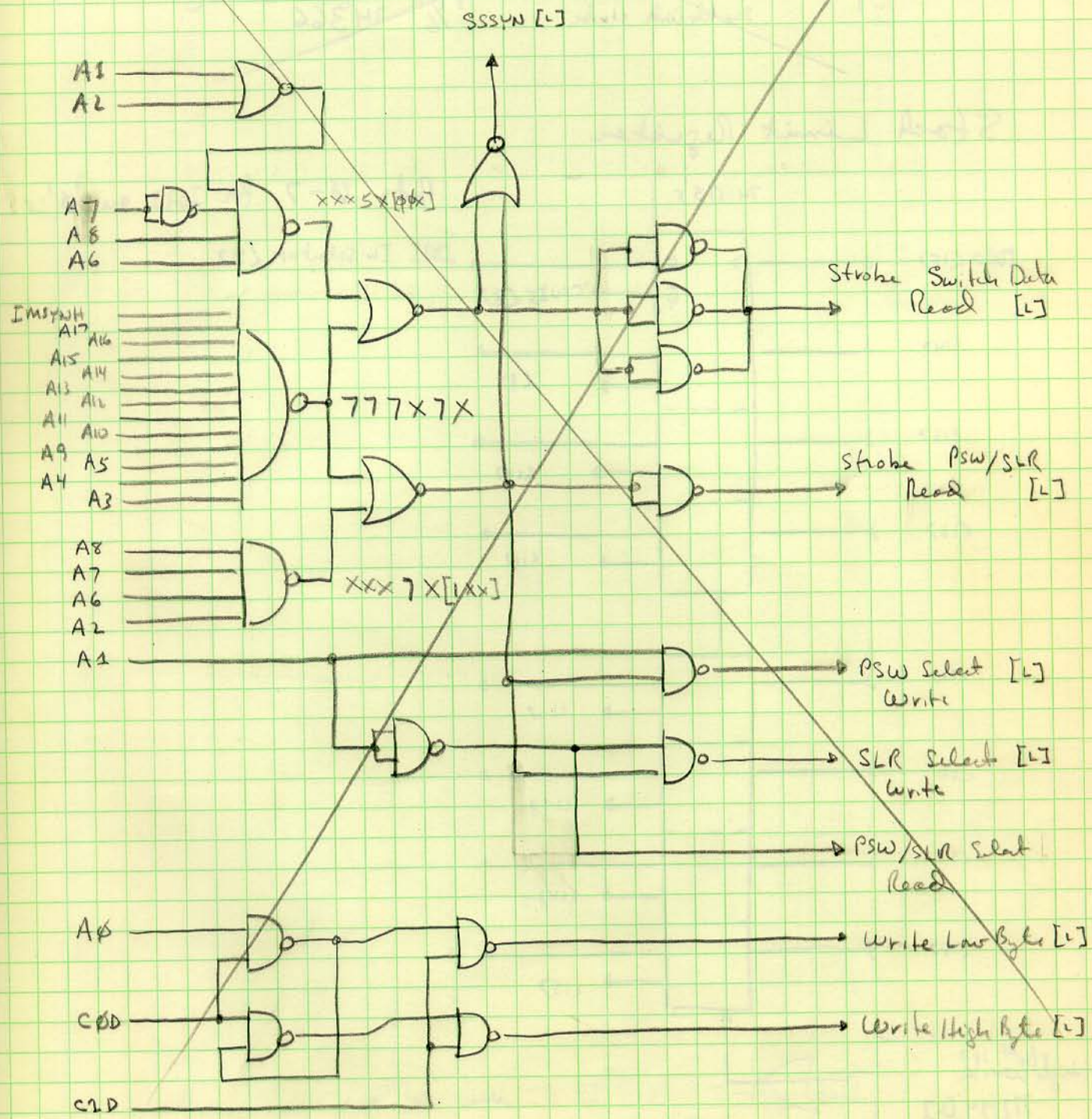
✓	8	<63:70>	I/O BUS CONTROL (MMGT coding)
✓	12	<71:82>	Controller Branch Address
✓	3	<83:85>	Controller Operation
✓	5	<86:90>	Branch Conditions

5 lines remaining for 96 Bit Microcode

✓	1	<91>	PSW / Constant Select
✓	1	<92>	extra bit in Bus control explicit immediate

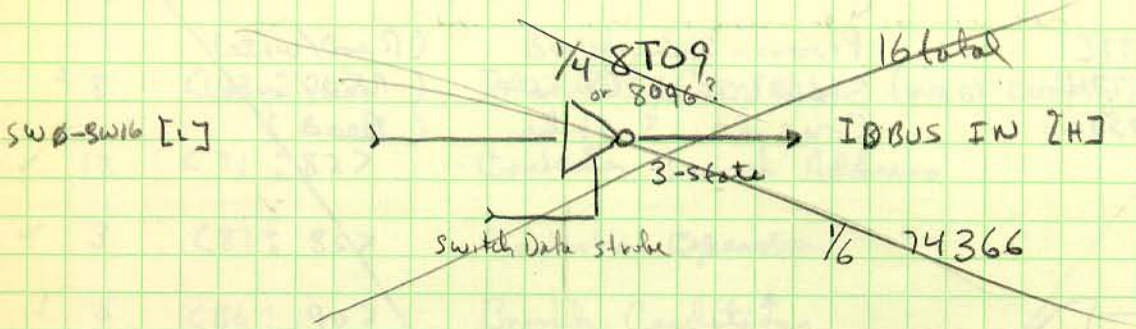
Preliminary BUS Address Decoder for .

777776	Processor Status Word	(Read/write)
777774	Stack Limit Register	(Read/write)
777570	Console Switches	(Read)

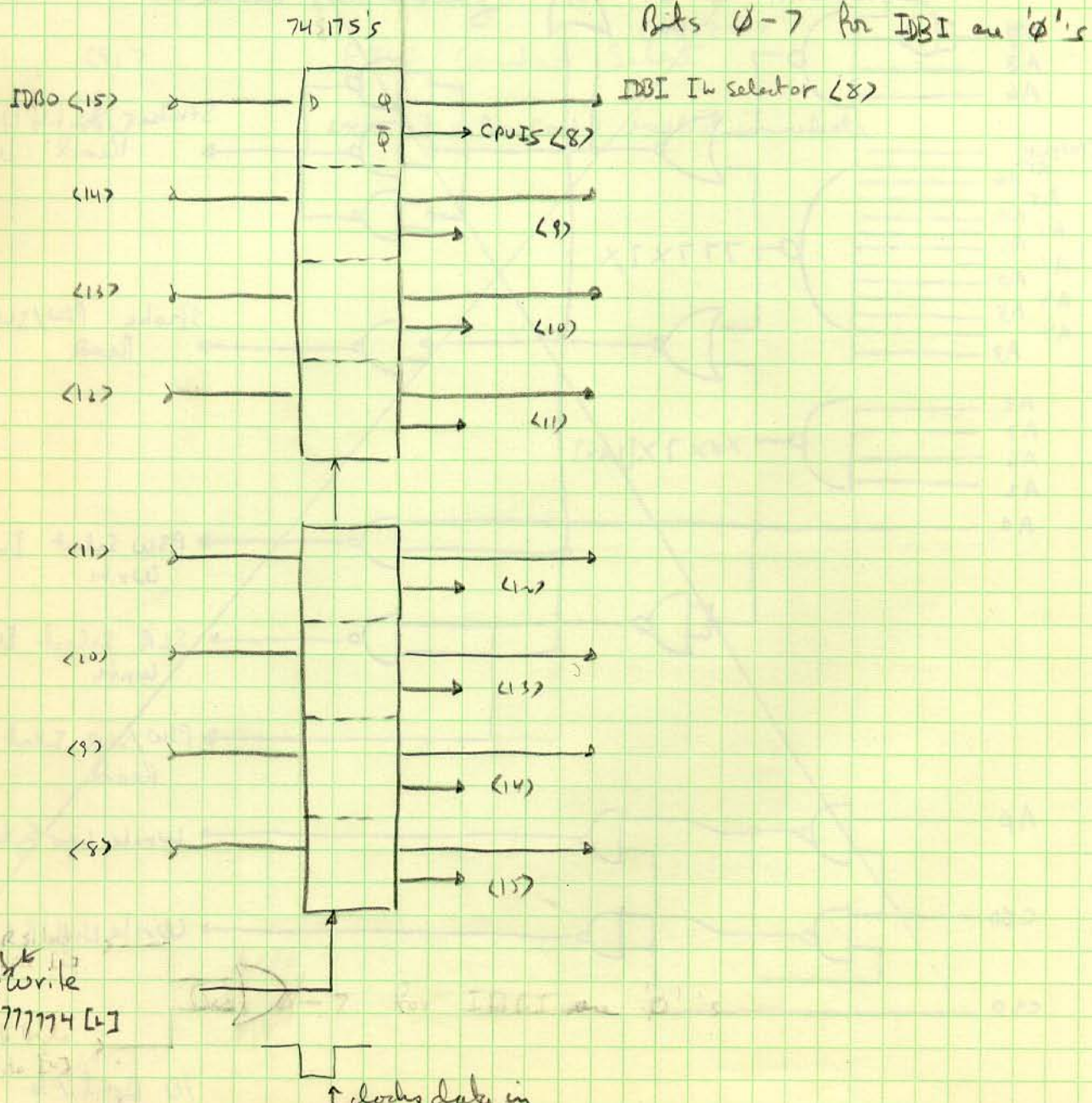


10 April 76
ARD

Console Switch Register Data strobe



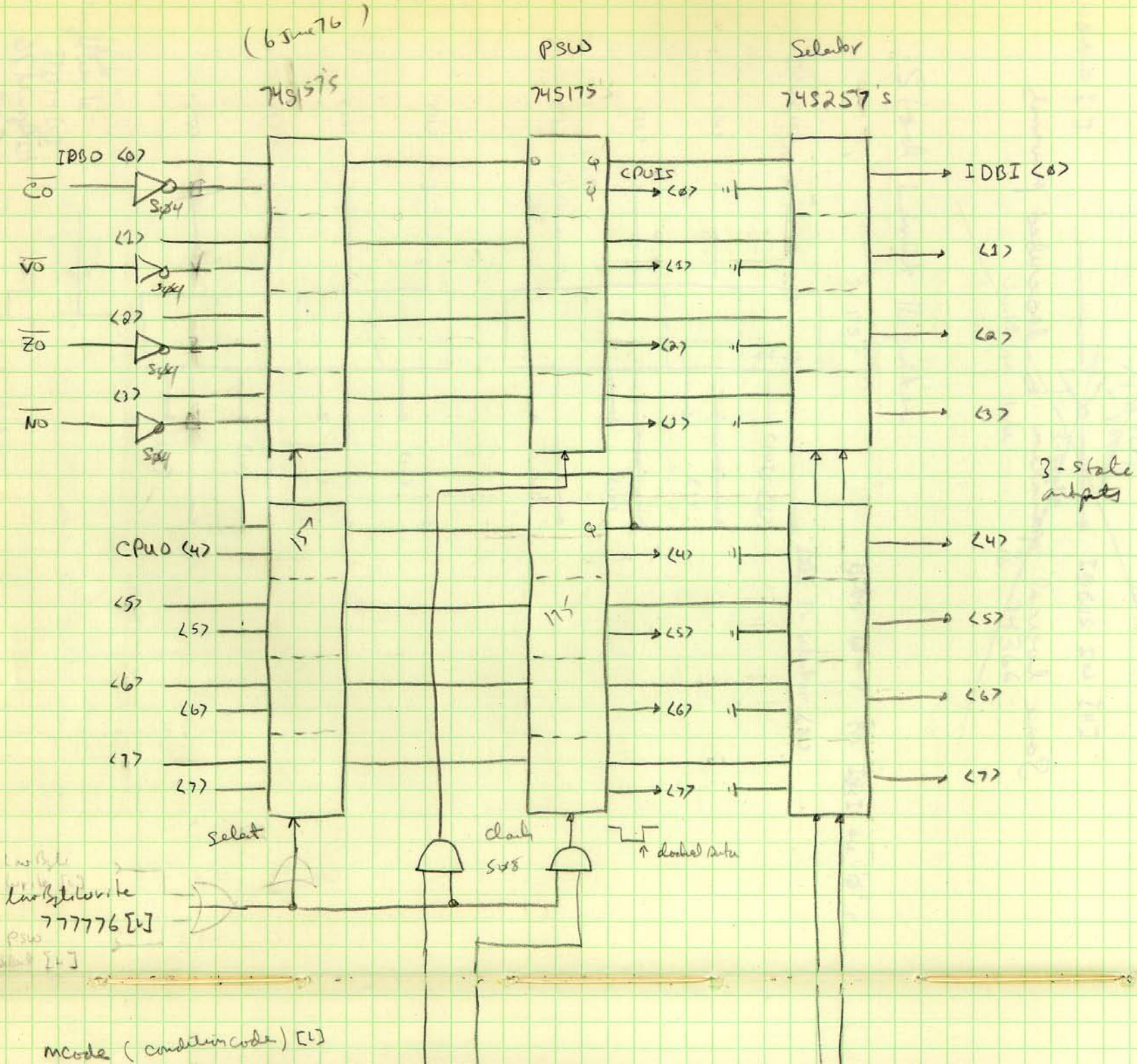
Stack Limit Register

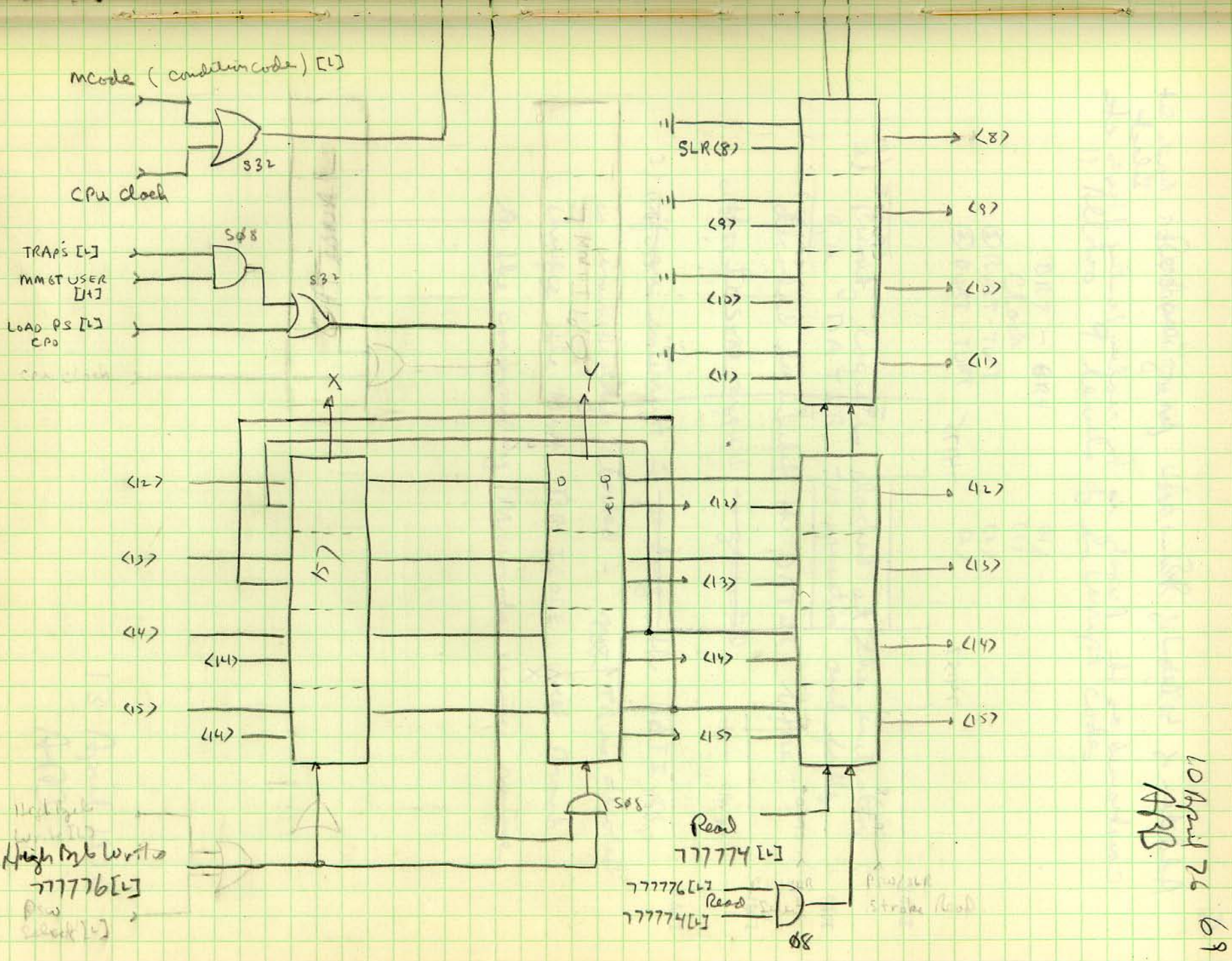


Preliminary Revision of the PSW logic

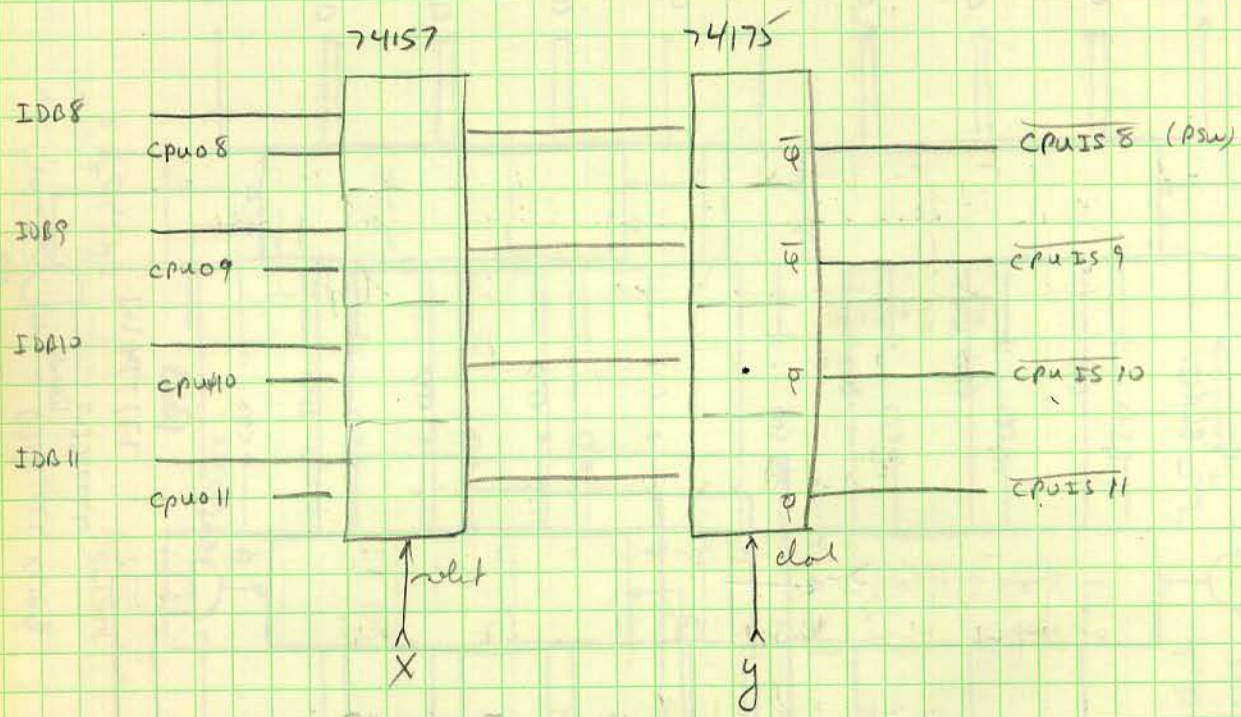
Some logical operation as described previously

10 April 76
AKD


(6 sure 76)
74S175's
PSW
74S175
Selector
74S257's
IDBO <0>
CO
SØ4
<1>
VO
SØ4
<2>
ZO
SØ4
<3>
NO
SØ4
CPUIS
<0>
<1>
<2>
<3>
IDBI <0>
<1>
<2>
<3>
3-state outputs
CPUO <4>
<5>
<6>
<7>
15
17
<4>
<5>
<6>
<7>
<4>
<5>
<6>
<7>
Selat
clock
SØ8
doubled rate
LowByteWrite
777776 [4]
PSW
Select [4]
MCode (condition code) [4]


MCode (condition code) [L]
S32
CPU clock
TRAPS [L]
MMGT USER [L]
LOAD PS [L] CPD
S08
S37
SLR(8)
<8>
<9>
<10>
<11>
<12>
<13>
<14>
<15>
X
Y
D
Q
57
S08
Read
777774 [L]
777776 [L]
Read
08
High Byte Write
777776[L]
10 April 76
ARD
69

Addition of 4 Bits of Kernel/User MMGT PAR/PDR/CSR Register Set Select
Constrained as the Priority is for reading/writing/traps etc



Redesign of the BR Logic

1) Allows 9 levels of interrupt codes

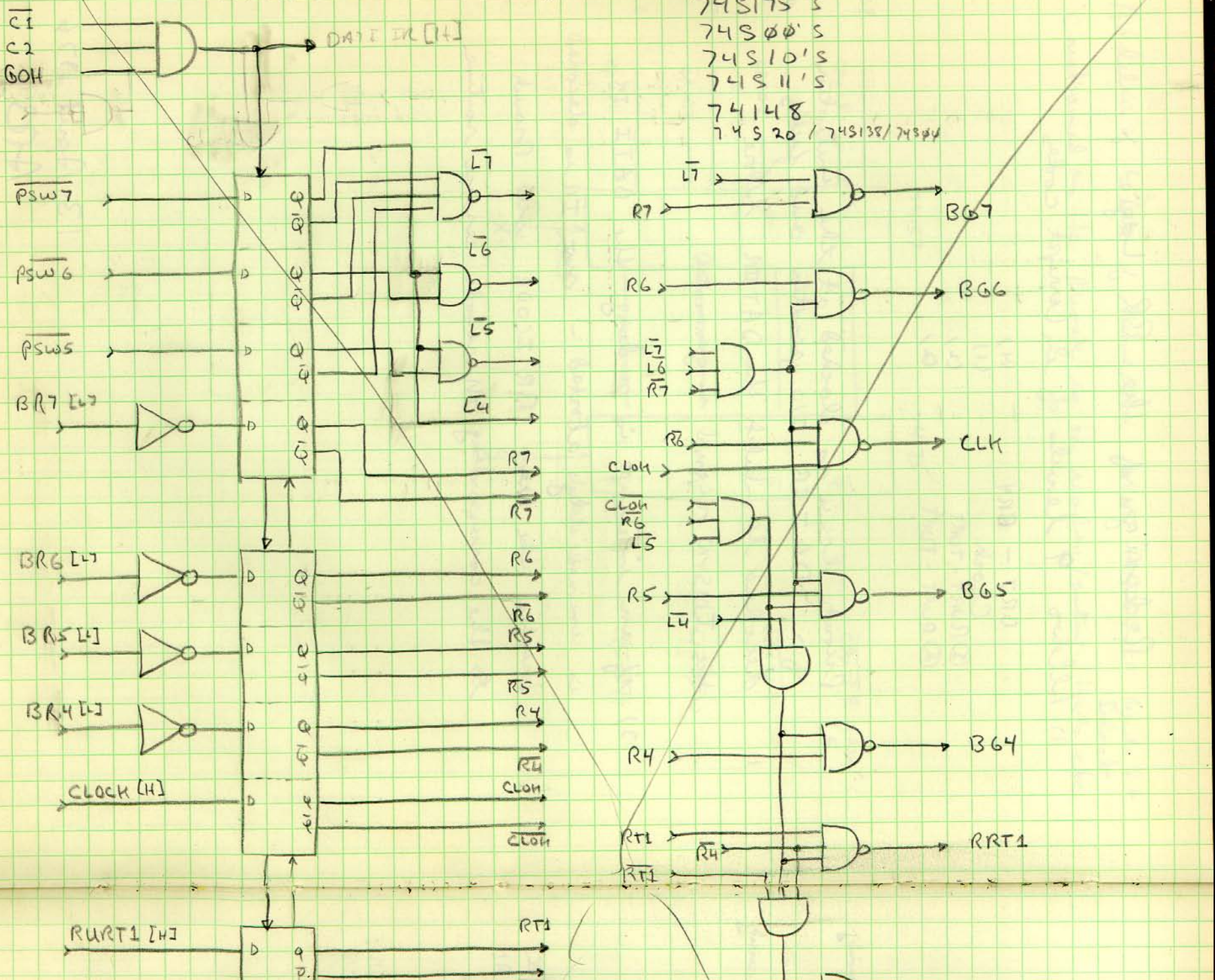
BR7 – BR4	(4)
Clock	(1)
① RURT - TURT	(2)
② RURT - TURT	(2)

2) Priority logic is clocked at the initiation of a DATIR I/O operation and fully decoded to inhibit the DATIR before the IMSYN signal is generated

3) if an interrupt is pending the DATI IR is immediately aborted – no IR is decoded therefore the Next BR ICODE will Branch to the corresponding Microcode service routine

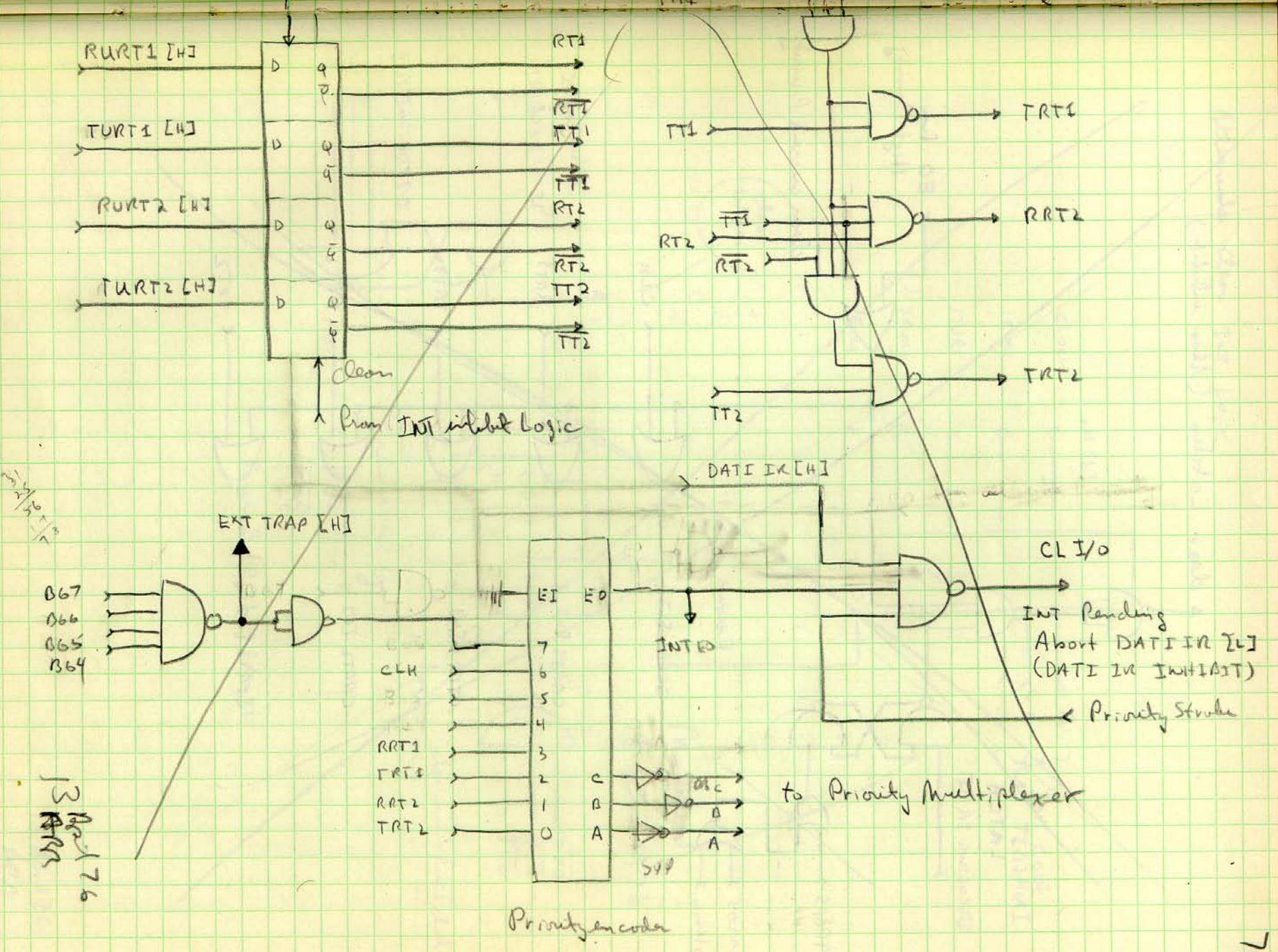
13 April 1976
APB

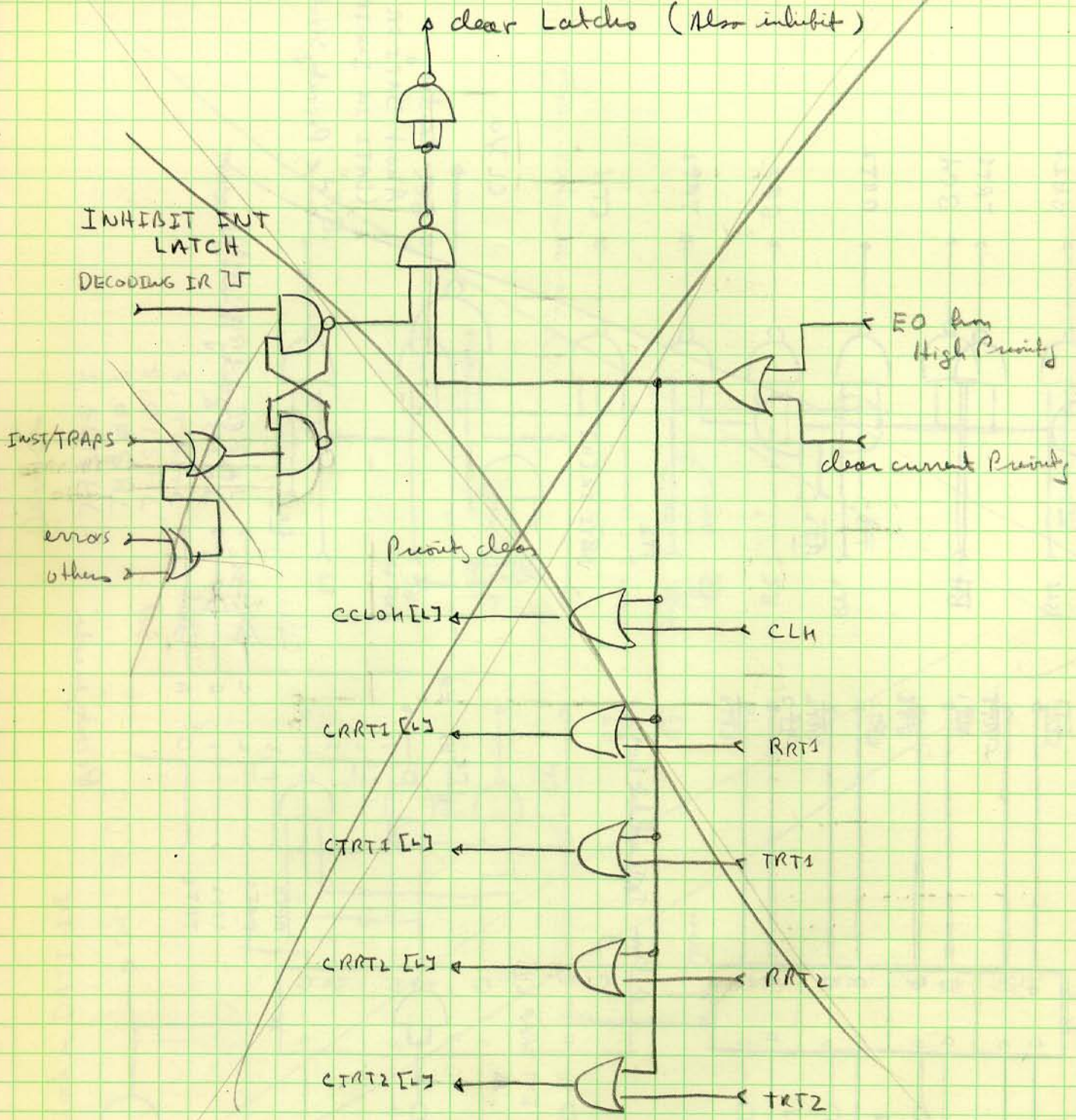
Clock on DATI IR



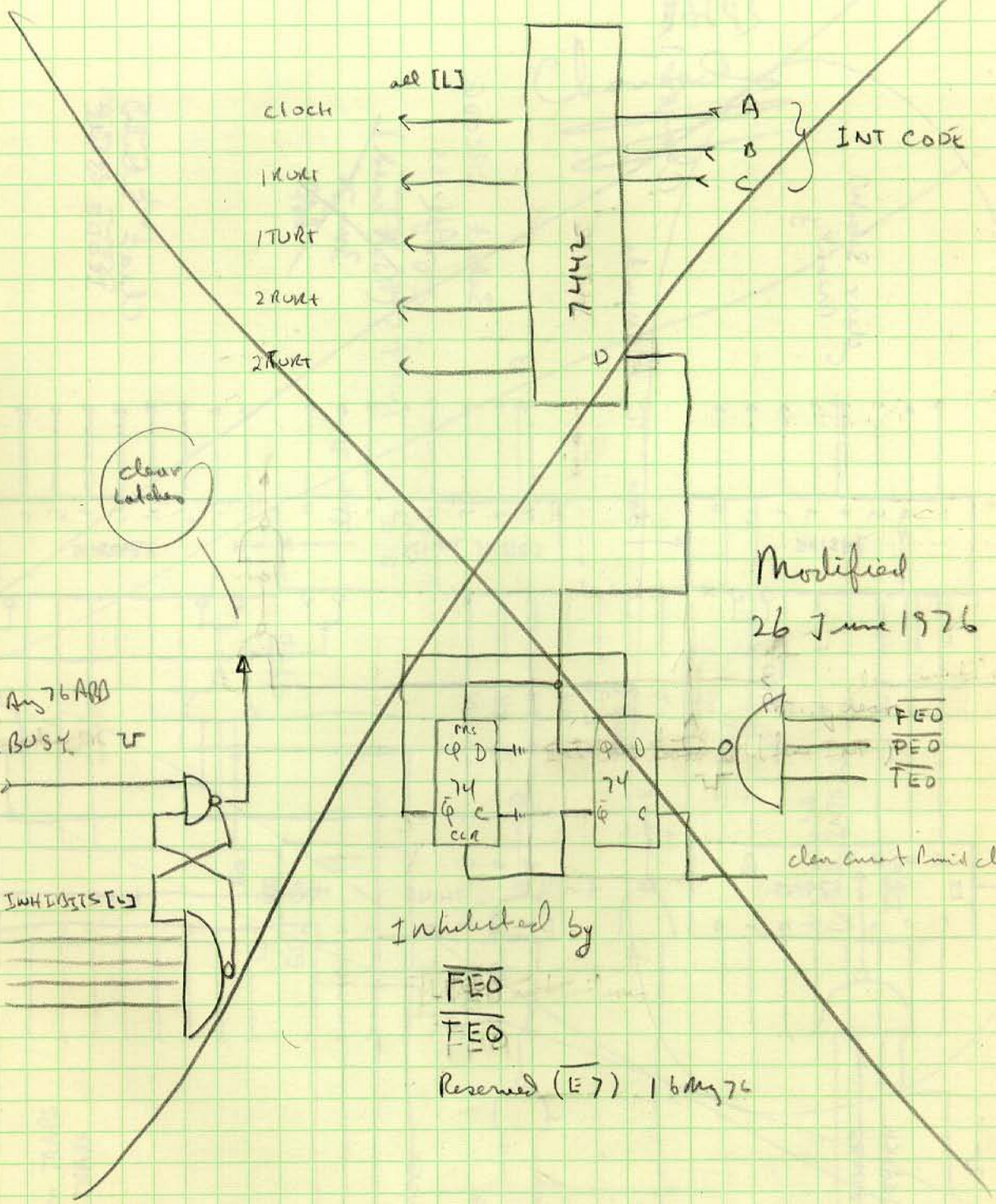
Devices

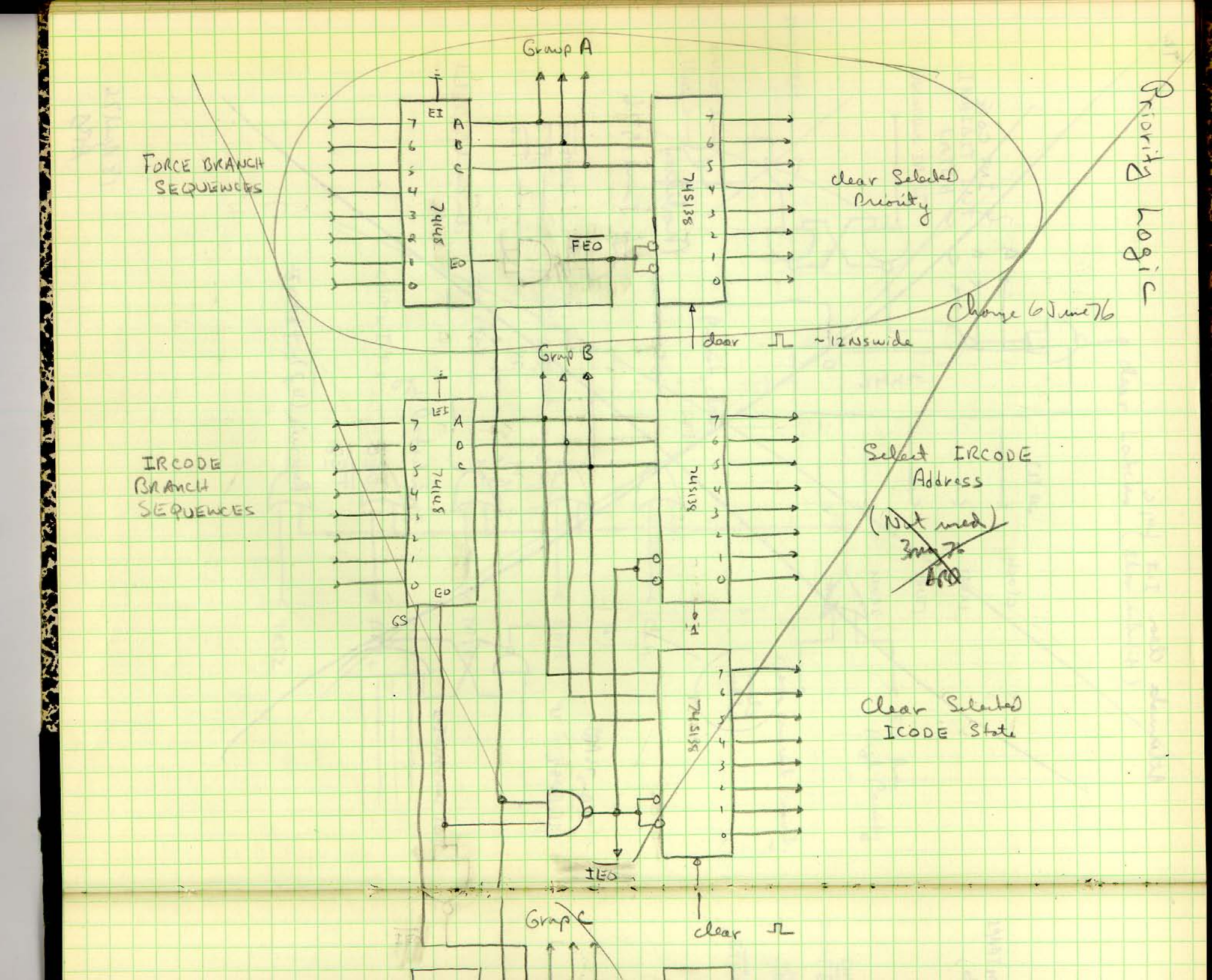
74S175's
74S00's
74S10's
74S11's
74148
74S20 / 74S138 / 74S64

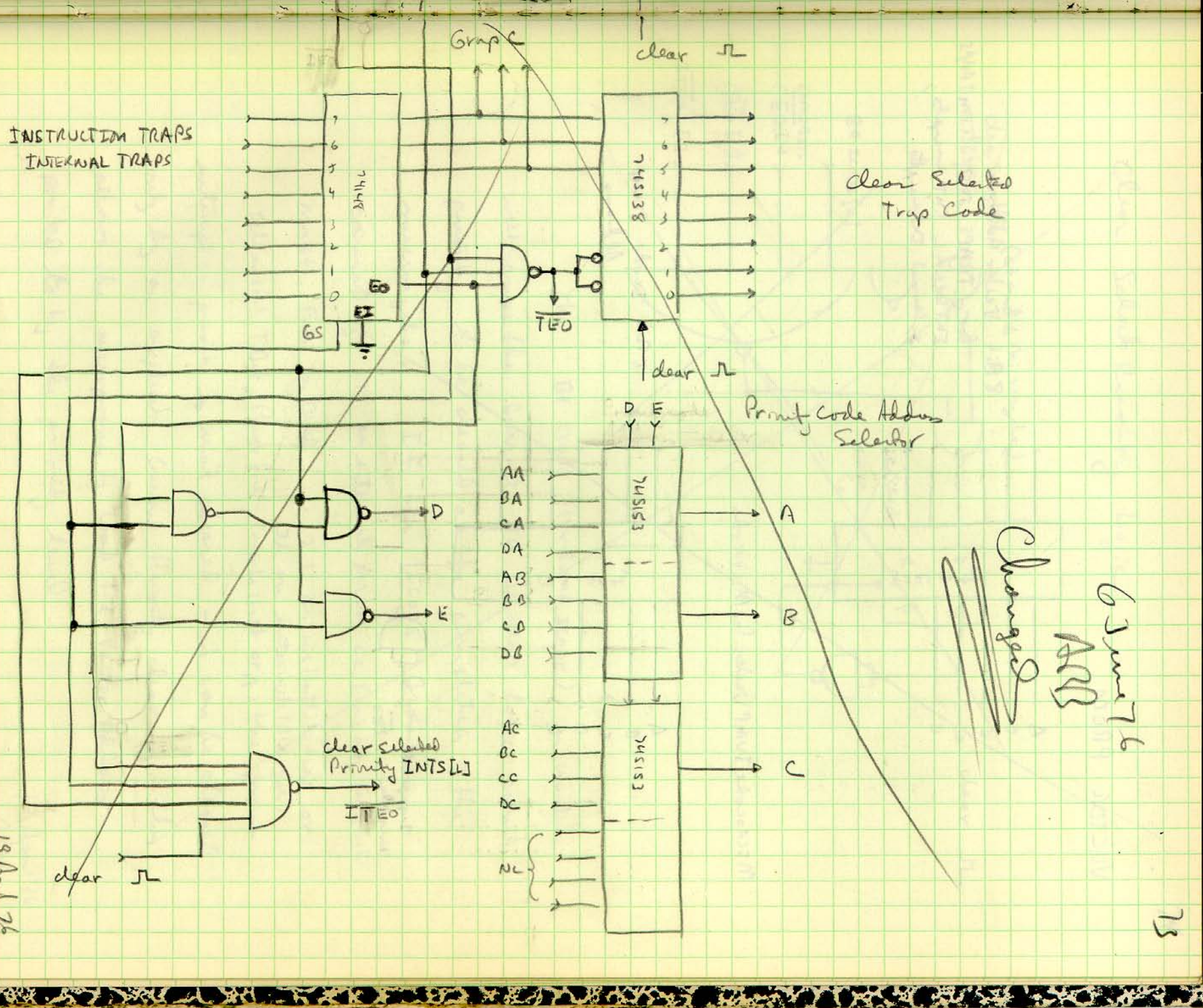

RURT1 [H]
TURT1 [H]
RURT2 [H]
TURT2 [H]
D
Q
RT1
TT1
RT2
TT2
clear
from INT inhibit Logic
TRT1
RRT2
TRT2
DATI IR [H]
EXT TRAP [H]
B07
B06
B05
B04
CLK
RRT1
TRT1
RRT2
TRT2
EI
EO
INTEO
CL I/O
INT Pending
Abort DATI IR [L]
(DATI IR INHIBIT)
Priority Strobe
to Priority Multiplexer
Priority encoder


clear Latches (also inhibit)
INHIBIT INT LATCH
DECODING IR [L]
INST/TRAPS
errors
other
EO from High Priority
clear current Priority
Priority clear
CCLOH [L]
CLH
CRRT1 [L]
RRT1
CTRT1 [L]
TRT1
CRRT2 [L]
RRT2
CTRT2 [L]
TRT2

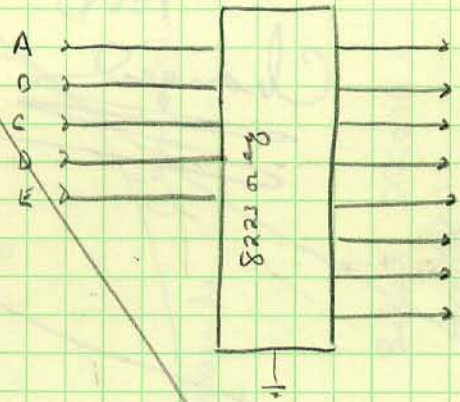
Alternate clear INT Logic




Group A
FORCE BRANCH SEQUENCES
clear Selected Priority
Priority Logic
Change 60 ns to 76
clear ~12NS wide
Group B
IRCODE BRANCH SEQUENCES
Select IRCODE Address
(Not used)
Clear Selected ICODE State
Group C
clear

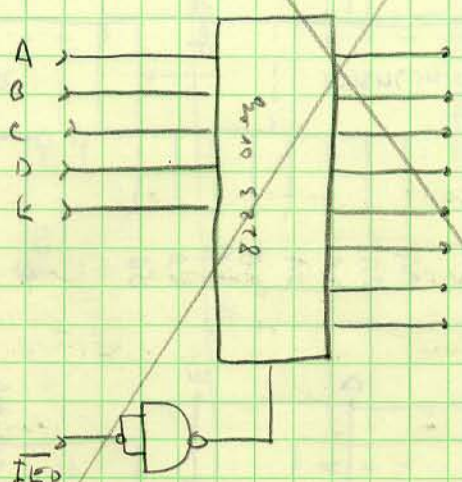

INSTRUCTION TRAPS
INTERNAL TRAPS
Group
clear
74148
74138
EO
EI
GS
TEO
clear
clear Selected Trap Code
D
E
Priority Code Address Selector
AA
BA
CA
DA
AB
BB
CB
DB
AC
BC
CC
DC
NC
74151
74151
74151
A
B
C
D
E
clear Selected Priority INTS[L]
ITEO
clear
Changed
6 June 76
ARB
15 April 76
ARB
75

VECTOR PROM



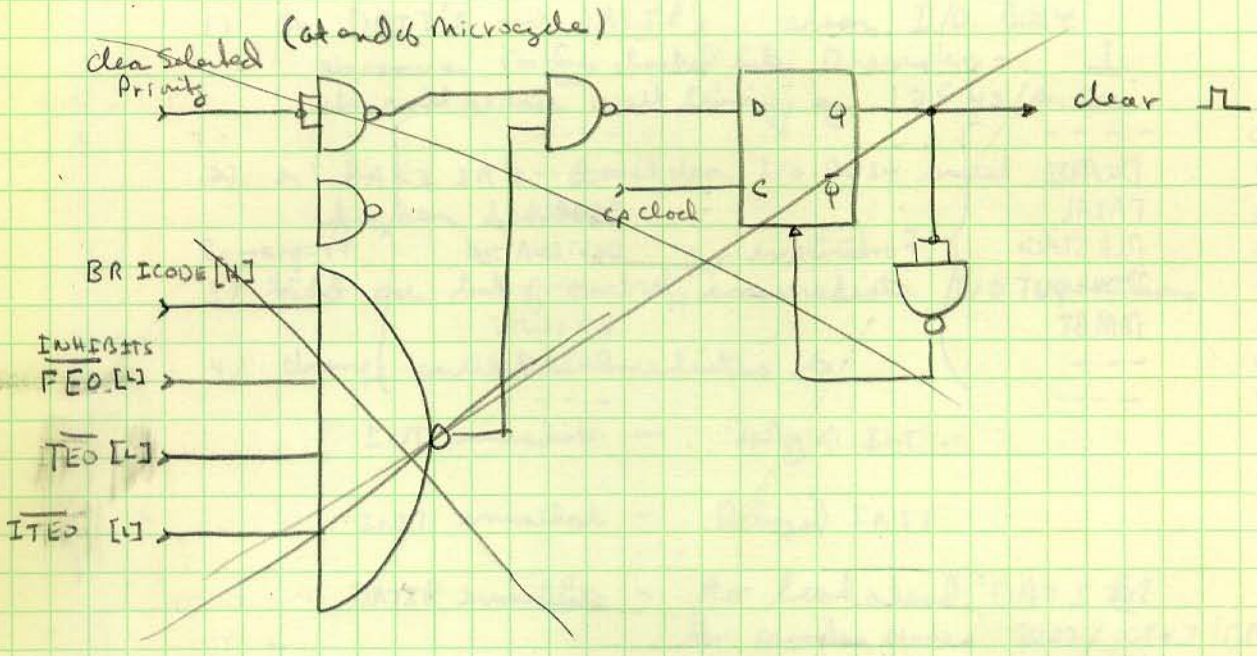
8 Bit Vector Address
for TRAPS / INSTRUCTION TRAPS
INTERNAL Interrupts,
External Interrupts

MICROCODE JUMP Vectors for Microcode



8 Bit Icode JUMP
Address

Clear Select Priority Logic



The BR ICODE (or JUMP ICODE) instruction will cause the current IR Decoded logic branch priority to be cleared at the completion of the microcycle. If the ICODE Branch is caused by 1) an internal error ie Emiac, Red stack Mmot, Fatal, or Dead or 2) an instruction trap or 3) an interrupt (BUS type) Yellow stack, T Bit, Power fail, Console Halt or ∧ the current interrupt priority is not cleared - it can be cleared only by an explicit Current Priority clear - this allows the microprogram to load the Vector Address pointed to by the priority level.

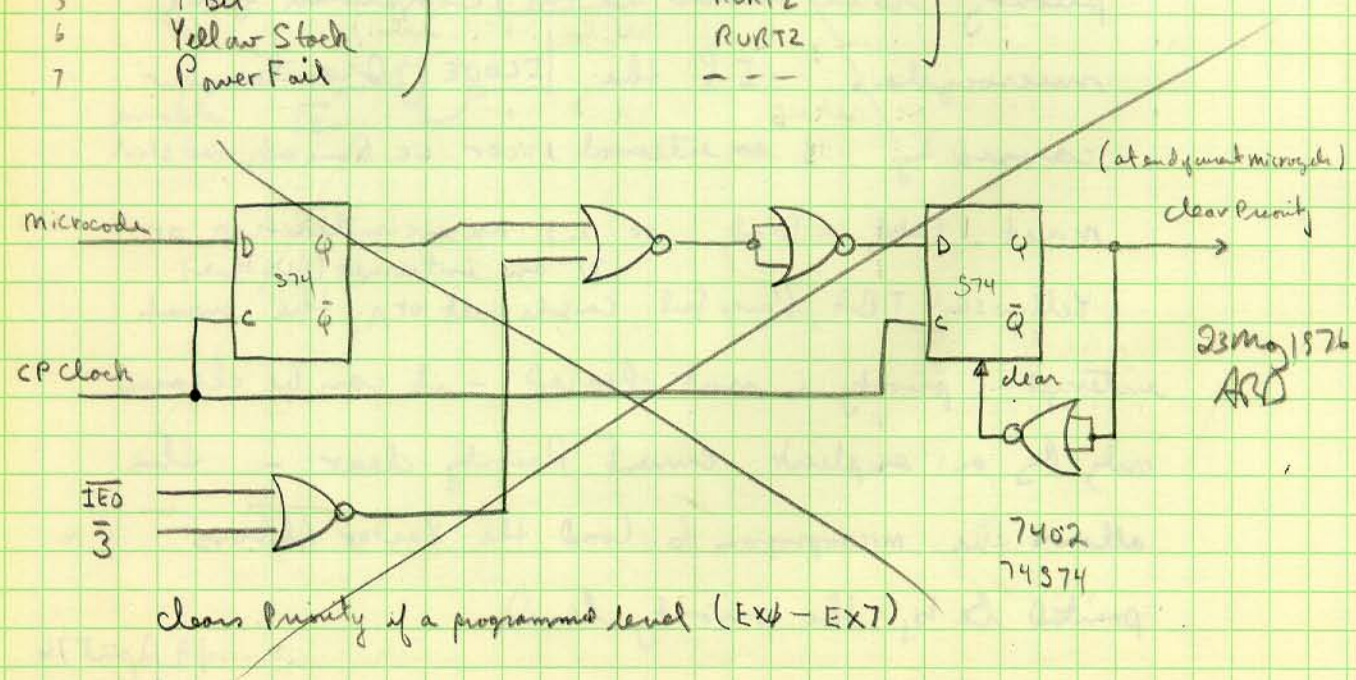
19 April 76
ARB

the priorities are reorganized as follows

Groups 1-4

	1		2	
0	- - - -	Force	- - - -	Programmed IR
1	DEAD		- - - - -	
2	FATAL		SOURCE	
3	Red STACH		DESTINATION	
4	TIME OUT		- - - -	
5	MMGT		EXECUTE	
6	- - -		RESERVED	
7	- - - -		- - - -	

	3		4	
0	EMT	Traps	- - -	Interrupts
1	TRAP		BG4-BG7	
2	IOT		CLOCK	
3	BPT		RURT1	
4	CONSOLE HALT		TURT1	
5	T Bit		RURT2	
6	Yellow Stack		RURT2	
7	Power Fail		- - -	



Changes Needed in the BUS control Logic

1) on DATI's and DATIP's clear I/O BUSY as soon as Data is clocked into Dregister - this will reduce wait timing by 125 NS/per I/O

2) on DATI IR's dont clear I/O BUSY until IR has been decoded

3) add one bit of control for explicit immediate I/O operations

4) change current immediates to

INT immediate — Delayed INT

INIT immediate — Delayed INIT

DATIR immediate - to lookahead DATIIR for ~~Modes~~ Modes SS[0] / DD[0] IBR

5) on DATO's and DATOB's clear I/O BUSY as soon as SSYN is received decrease wait timing by 125 NS/per I/O

approximate I/O timing (add 150NS for MMGT Enabled)

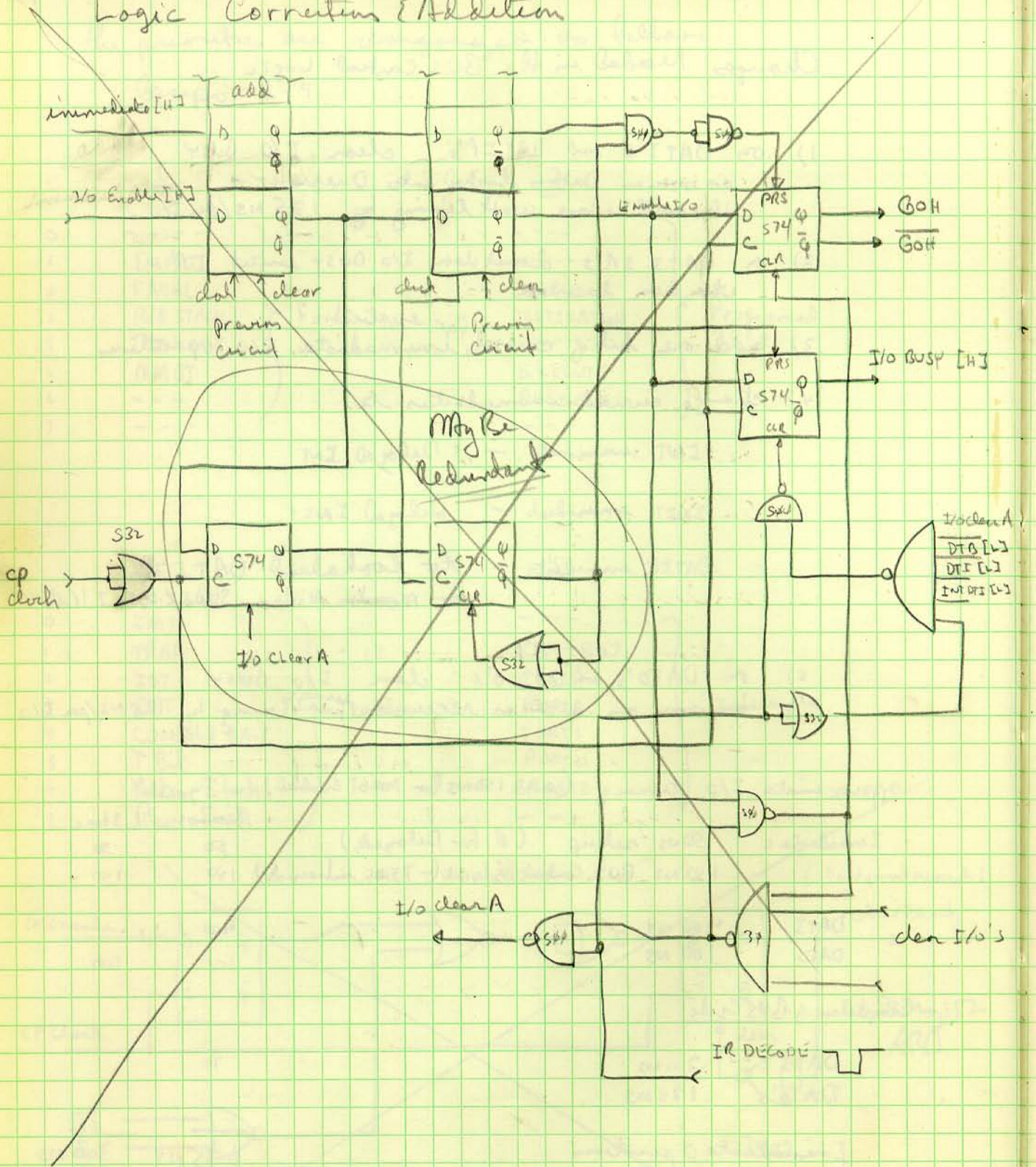
		Typical Access	Store
Immediate's	50NS setup (0 for Delayed)	50	50
	100 NS BUS control (External (-75NS internal)	100	100
DATI	400 NS	400	
DATO	100 NS		100

Completion BUS Cycle

DATI's	200 NS	75
DATO's	125 NS	
Immediate operations		675 NS / 300 NS
Delayed Operations		625 NS / 250 NS
Internal BUS		550 NS / 175 NS
DATI IR's (Internal Bus & Immediate)		~ 800 NS

21 April 76

Logic Correction / Addition



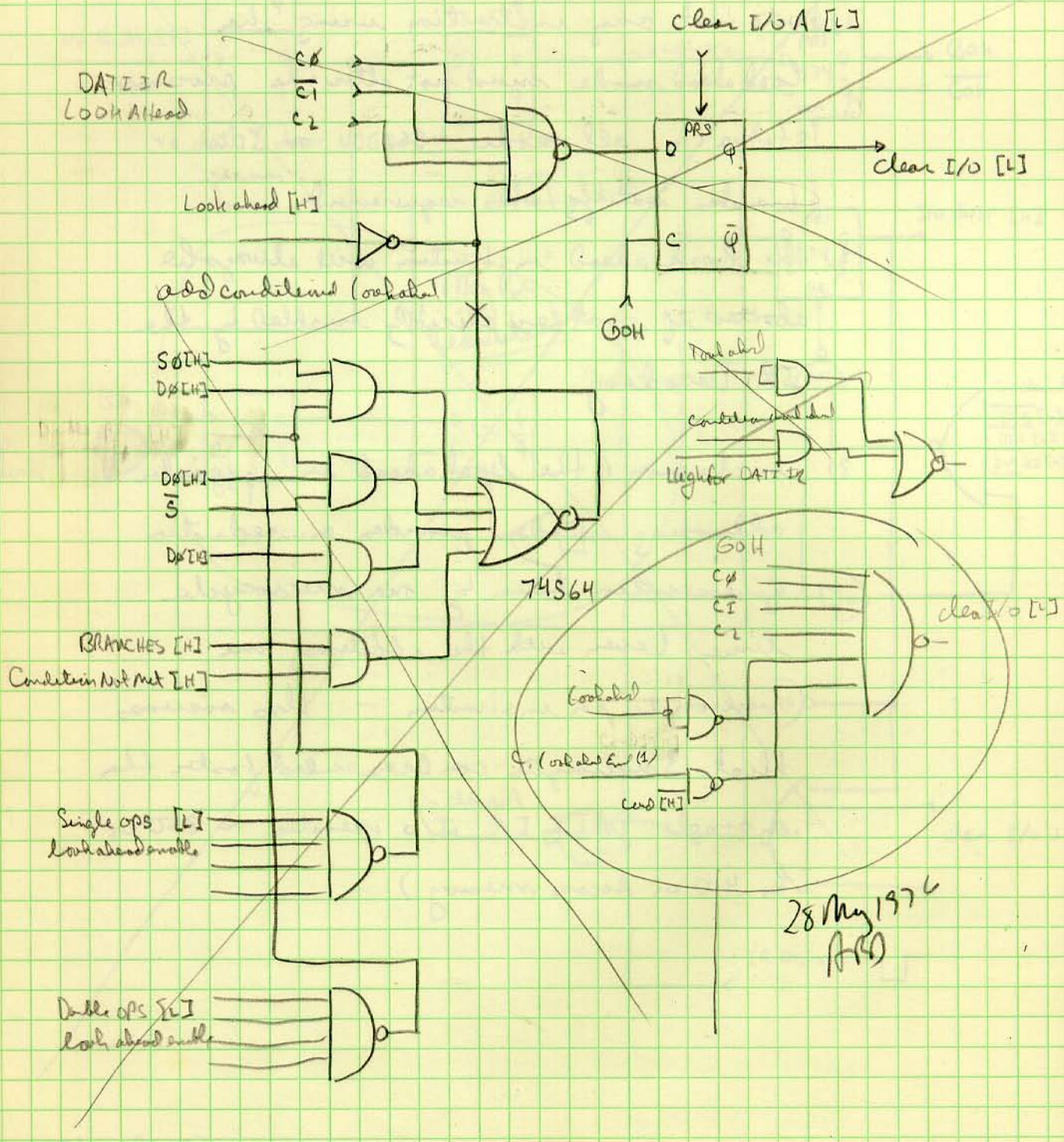
23 April 76
ASD

Notes on the Look ahead DATI IR

1) Note that any instruction using the lookahead mode must not alter the processor status - all mode SS@DD@ or SDD@ or branches satisfy this requirement.
2) the look ahead instruction will always be aborted if not explicitly enabled by the IR Decoding
3) inclusion of the lookahead in register addressing modes provides a reduction of execution time by one microcycle time (even with the additional one microcycle per instruction - this assumes that 3 microcycles can be executed faster than a single DATI IR I/O operation ~ 850 ns for 400 ns Access memory)

23 April 76
ARB

DATI IR Lookahead Inhibit/enable



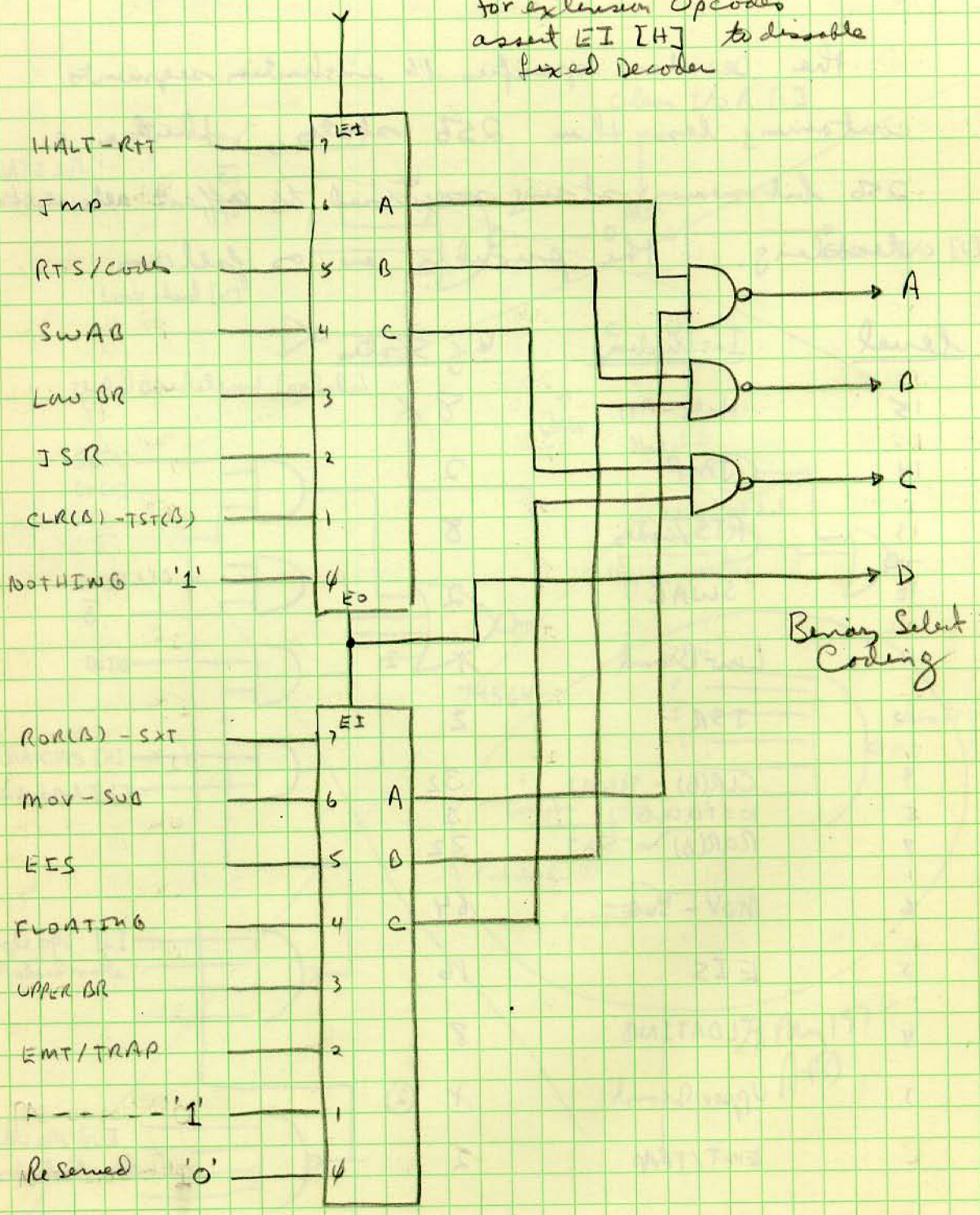
Preliminary Instruction Decoding

The decoder specifies 16 instruction segments containing less than 256 states, therefore a 256 bit memory string may be used to affect all instruction decoding. The priority is as follows

level	Instructions	# of States
15	Halt - RTI	8
14	JMP	2
13	RTS/codes	8
12	SWAB	2
11	Low Branch	X 2
10	JSR	2
9	CLR(B) - TST(B)	32
8	NOTHING	1
7	ROR(B) - SXT	32
6	MOV - SUB	64
5	EIS	16
4	FLOATING	8
3	Upper Branch	X (2)
2	EMT/TRAP	2
1	- - - -	- -
0	RESERVED	1

JMay 1976
ARD

Encoding Hardware



Selector Bits (Connections)

Priority		0	1	2	3	4	5	6	7	# of Slots
	Extensions	I6	I7	I8	I9	I10	I11	0	0	64
(6)	MOV-SUB	S0	D0	I12	I13	I14	I15	0 (1)	0 (0)	64
(9)	CLR(B)-TST(B)	D0	I6	I7	I8	I15	0	1 (0)	0 (1)	32
(7)	ROR(B)-SXT	D0	I6	I7	I8	I15	1	1 (0)	0 (1)	32
(5)	EIS	D0	I9	I10	I11	0	0	0 (1)	1 (1)	16
(4)	FLOATING	I3	I4	I5	0	1	0	0 (1)	1 (1)	8
(15)	HALT-RTI	I0	I1	I2	1	1	0	0 (1)	1 (1)	8
(13)	RTS/NOP	I3	I4	I5	0	0	1	0 (1)	1 (1)	8
(14)	JMP	D0	0	0	1	0	1	0 (1)	1 (1)	2
(12)	SWAB	D0	1	0	1	0	1	0 (1)	1 (1)	2
(16)	JSR	D0	0	1	1	0	1	0 (1)	1 (1)	2
(2)	EMT/TRAP	I8	1	1	1	0	1	0 (1)	1 (1)	2
(11)	LOW BR	BR	0	0	0	1	1	0 (1)	1 (1)	2
(3)	UPPER BR	BR	0	0	0	1	1	0 (1)	1 (1)	---
(0)	Reserved	0	1	0	0	1	1	0 (1)	1 (1)	1
(X)	Nothing	1	1	0	0	1	1	0 (1)	1 (1)	1

IX is the [High True] instruction register bit

D0, S0 is the [High True] Mode Zero Code

5 May 76
ARD

Corresponding Connections for 16 line Selectors

Line #	Output Bit # 0	1	2	3	4	5	6 (modify)	7 (modify)
0	0	1	0	0	1	1	0	1
1	UNSPECIFIED							
2	I8	1	1	1	0	1	0	1
3	BR	0	0	0	1	1	0	1
4	I3	I4	I5	0	1	0	0	1
5	D0	I9	I10	I11	0	0	0	1
6	S0	D0	I12	I13	I14	I15	0	0
7	D4	I6	I7	I8	I15	1	1	0
8	1	1	0	0	1	1	0	1
9	D0	I6	I7	I8	I15	0	1	0
10	D0	0	1	1	0	1	0	1
11	BR	0	0	0	1	1	0	1
12	D0	1	0	1	0	1	0	1
13	I3	I4	I5	0	0	1	0	1
14	D0	0	0	1	0	1	0	1
15	I0	I1	I2	1	1	0	0	1

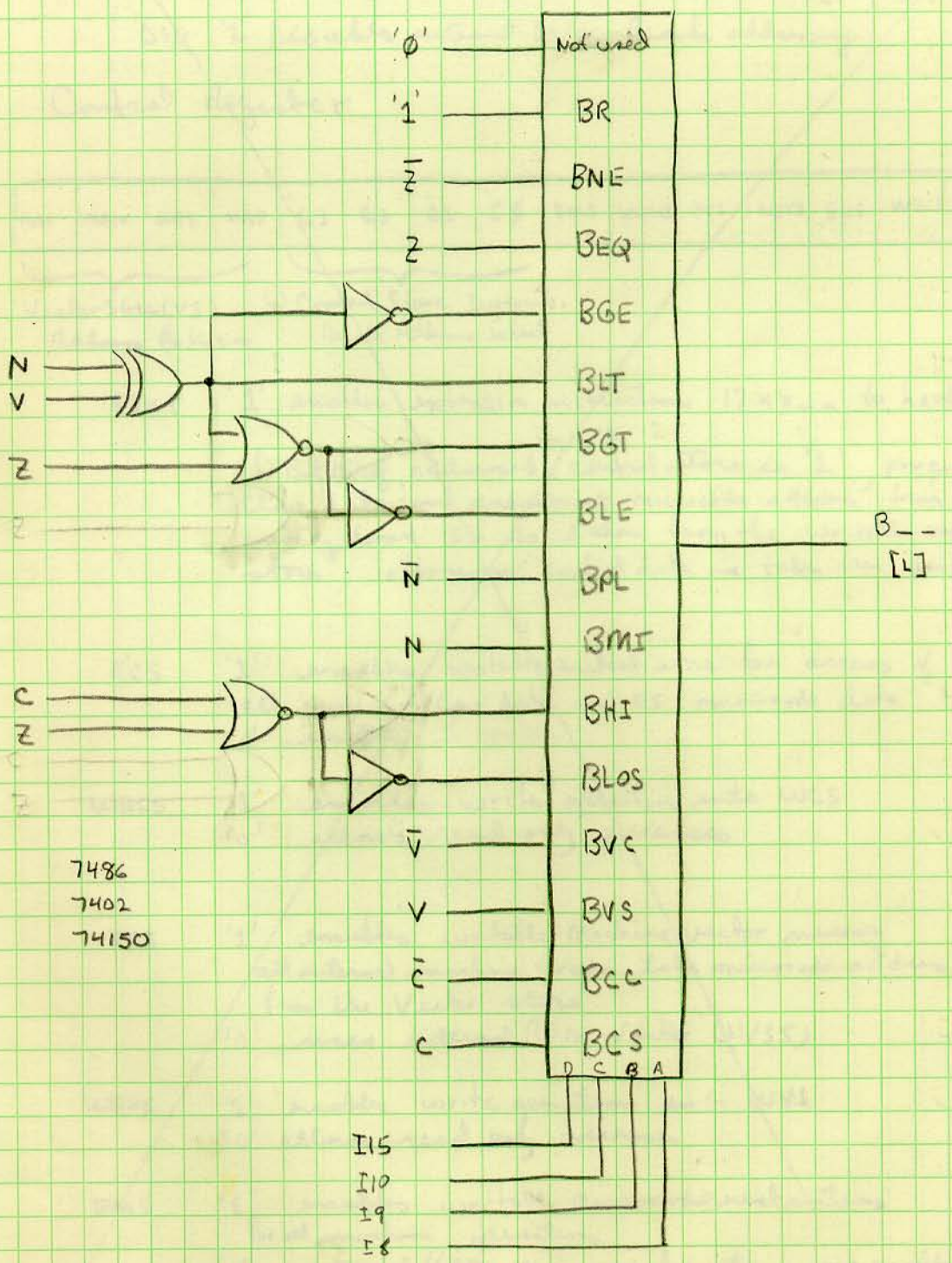
The 16 input selectors require low true signals

- '1' ⇨ [L]
- '0' ⇨ [H]

Ix ⇨ low true

D0, S0 ⇨ low true

Conditional BRanch INstruction Decoding



7486
7402
74150

9 May 1976
ARD

Writable Control Stores / Control Register

DSD '1' Disables internal DD/SS Branch addressing

Control Register

MA11	MA10	MA9	MA8	C3	C2	C1	C0	EMS	WEMS	EVS	WEVS	ECS	WECS	DSD	EEXT

Vector Store (VS) Address Extension

W Control Store Instruction Decode Address Select

EEXT '1' enables extension instructions 17xx__ to hardwired 11/35

if Bit 15 of addressed writable control store is '1', program flow data and subsequent microcode address from priority level ES is taken from the writable control store, otherwise control data is taken from hardwired store

ECS '1' enables writable control store for access of all program flow data and ES microcode data
'0' enables

WECS '1' enables write operation into WCS
'0' allows read only accesses

EVS '1' enables writable Machine vector memory all internal machine traps take microcode address from the Vector store
'0' uses internal ROM Vector (11/35)

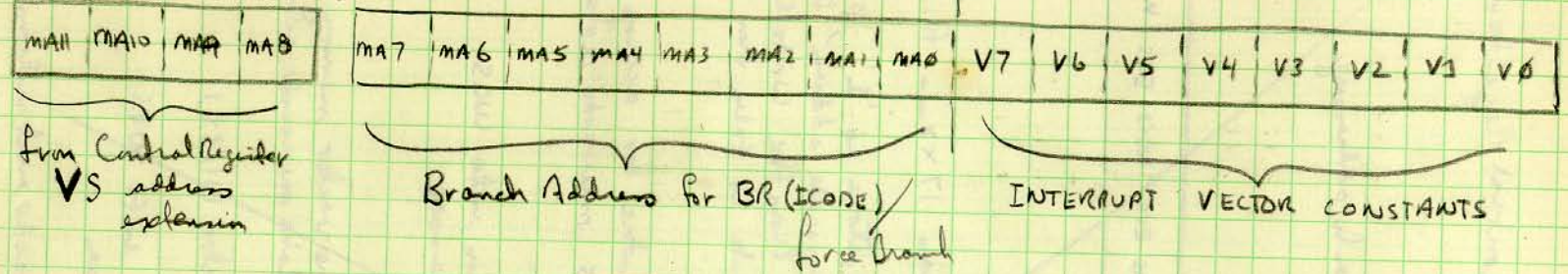
WEVS '1' enables write operations into WVS
'0' allows read only accesses

EMS '1' enables writable microcode instructions for all machine operations
'0' internal ROM Micro instructions are enabled

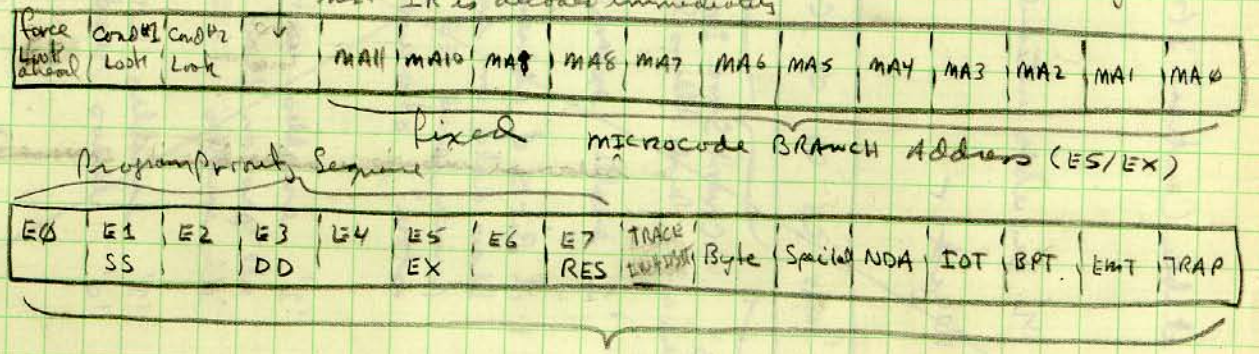
WEMS '1' enables write operations into WMS
'0' allows read operations only

10 May 76
ARB

Writable Vector Store (32 words only / one for each internal machine priority)



Writable Control Store (2 words per IR decode state / 2 words per microcode instruction)
with bit set, next DATI IR decode must wait until a WI/O L or WI/O H is asserted before decoding else next IR is decoded immediately



add conditional lookahead

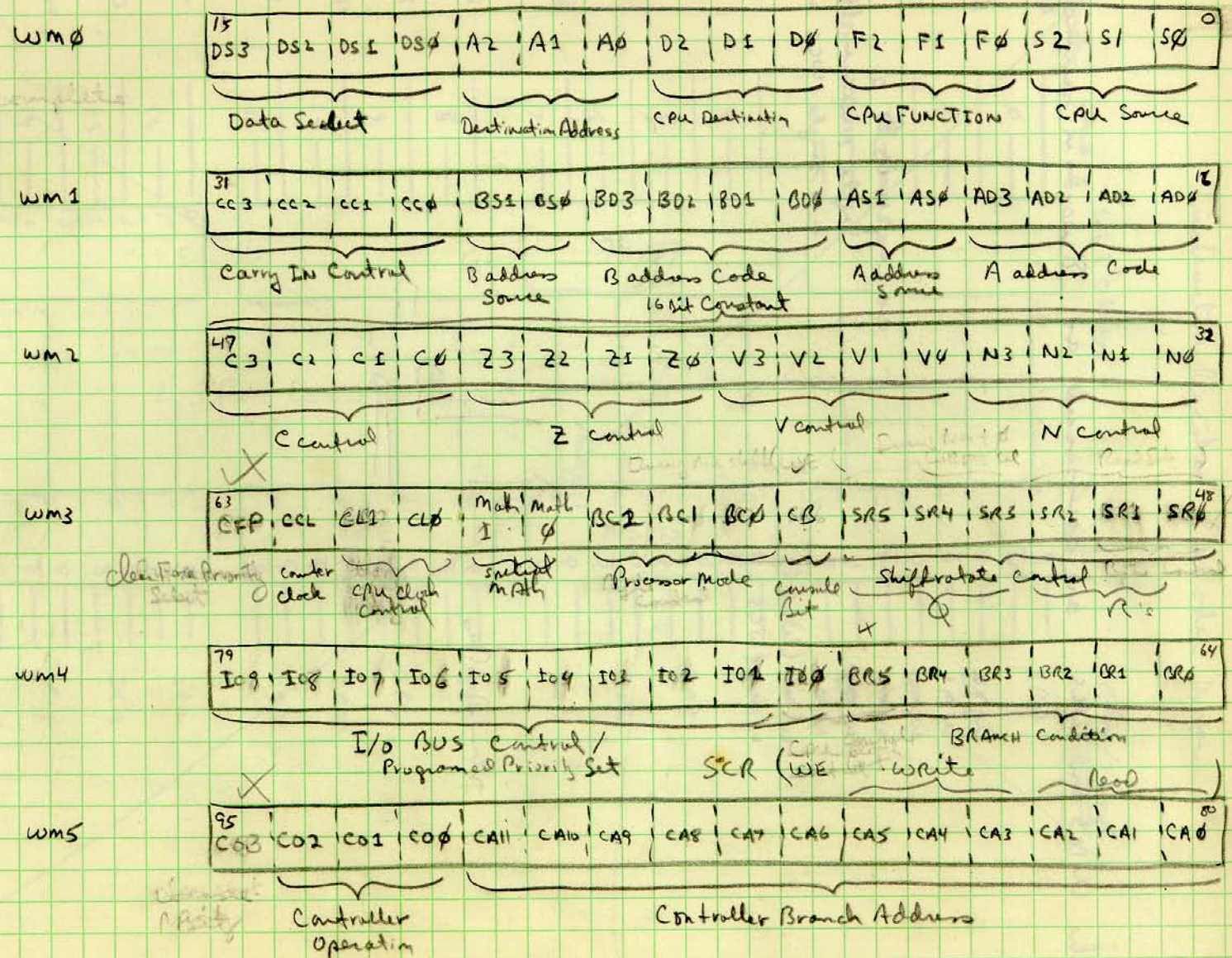
✓ Updated 27 Aug 76 AFB

✓ updated 6 Sept 76 AFB

✓ updated 21 Jan 77 AFB

Writable Microcode Store (6 words per microcode instruction (96 bit))

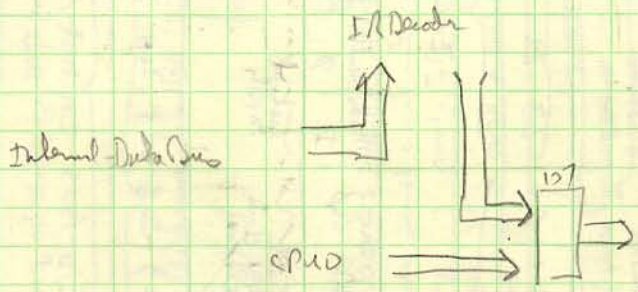
Writable Microcode Store (6 words per microcode instruction (96 bits))



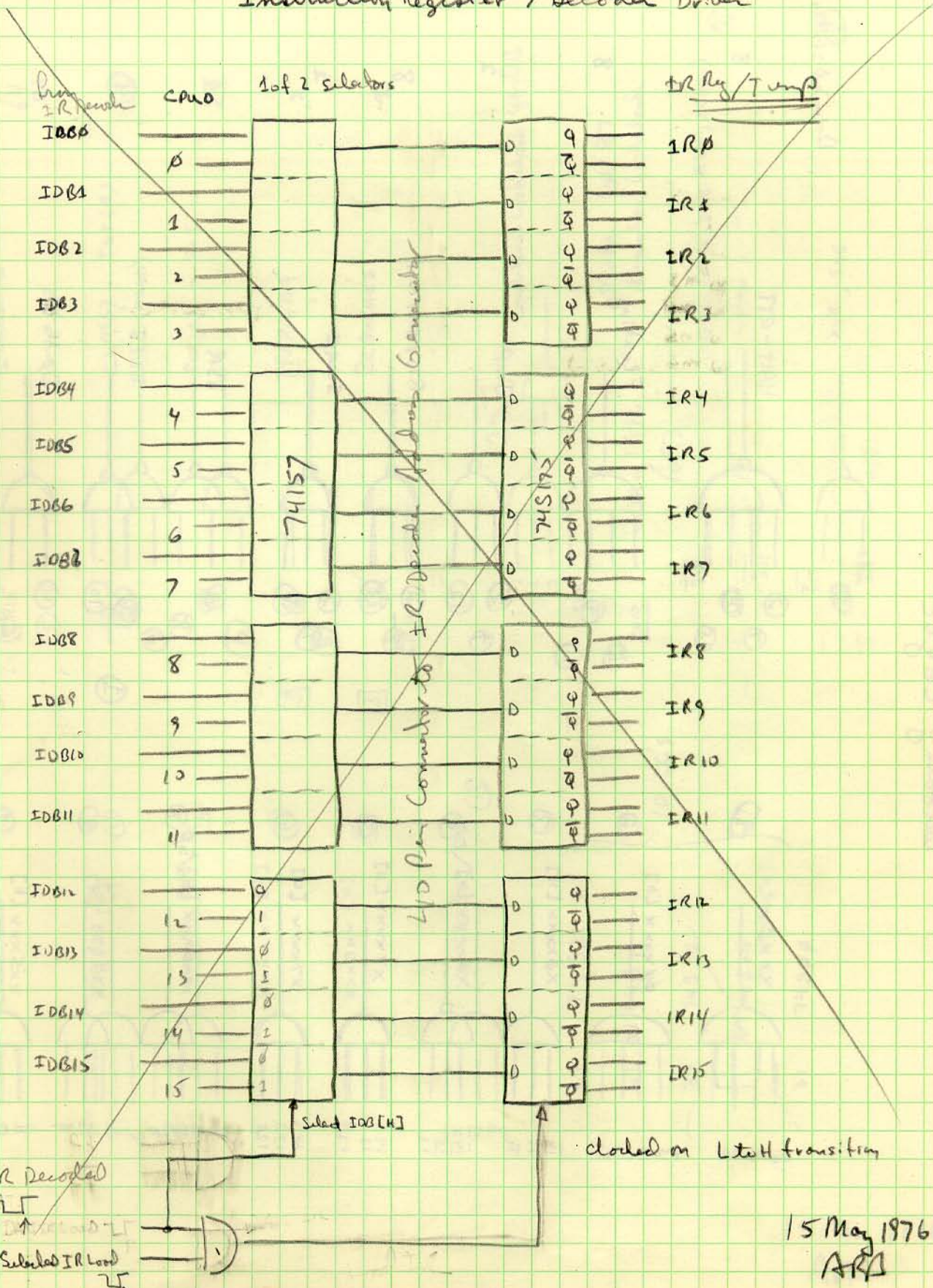
the WMS and WCS are accessed as 8 words in sequence

WM0
WM1
WM2
WM3
WM4
WM5
WM6 (WCS1)
WM7 (WCS2)

} corresponds to a
Single microinstruction
+ a single WCS instruction

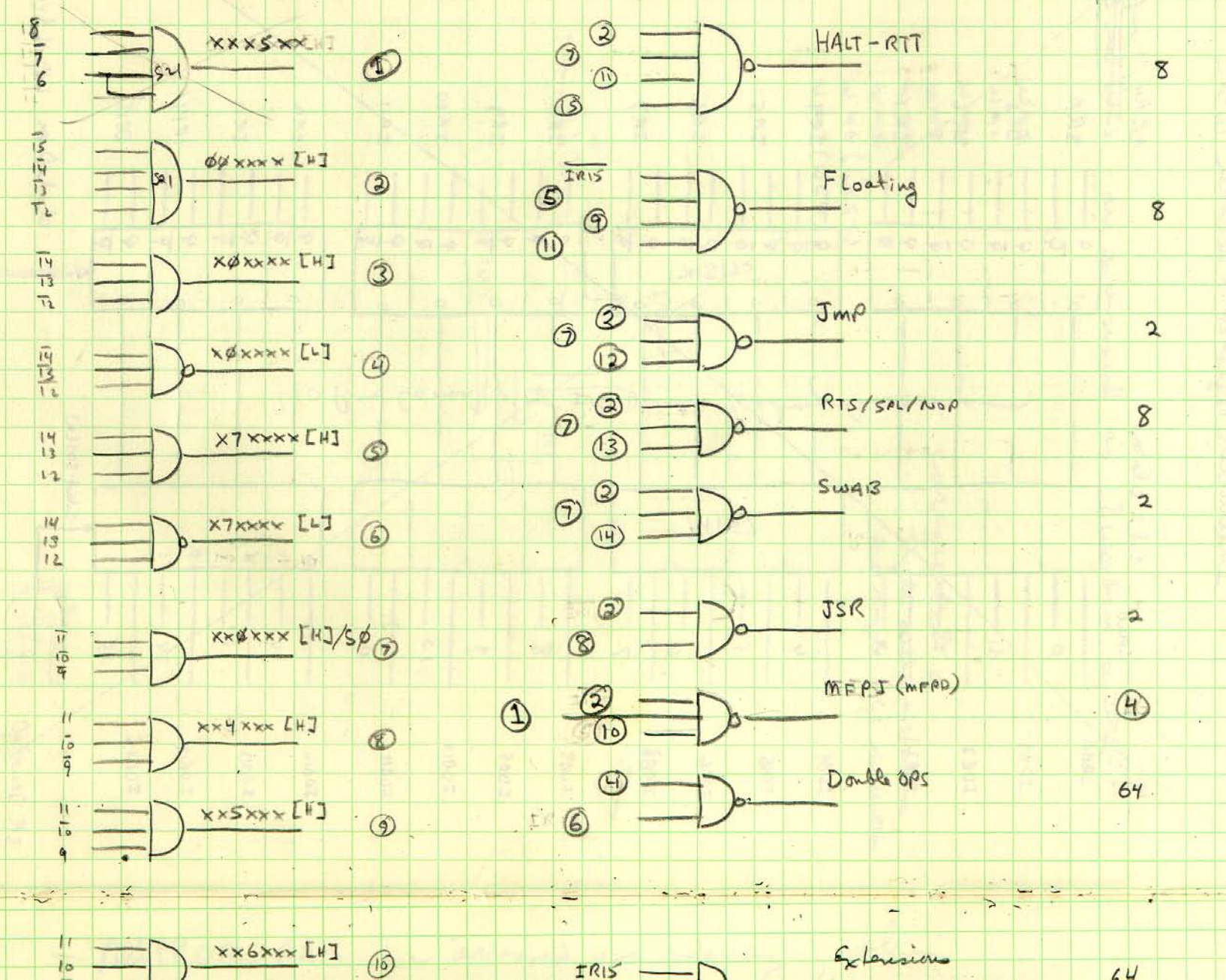


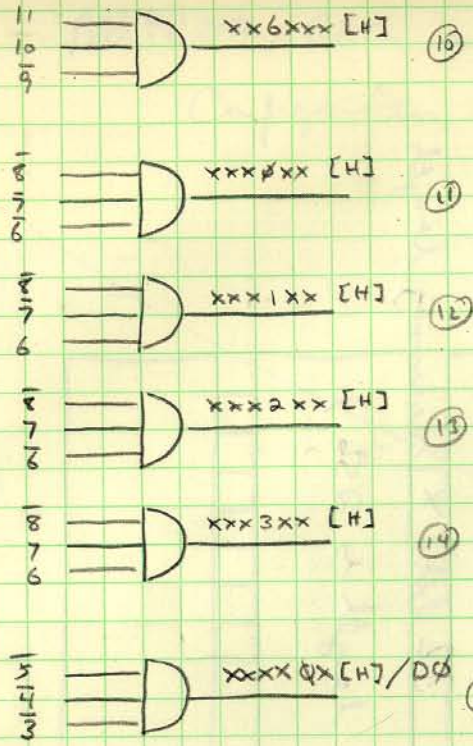
Instruction Register / Decoder Driver

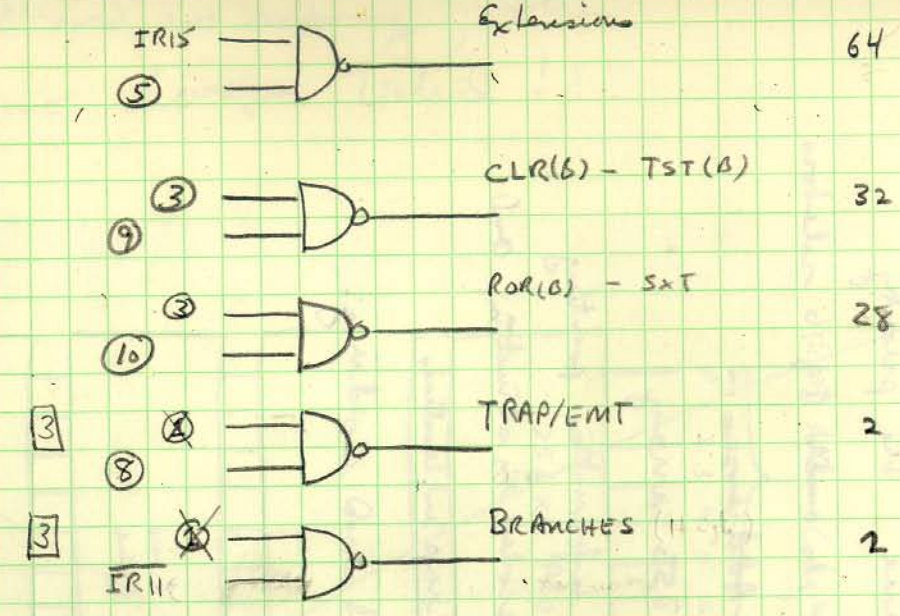


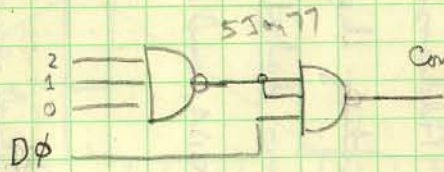
15 May 1976
ARM

Decoding Logic



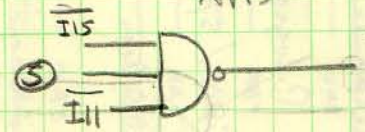


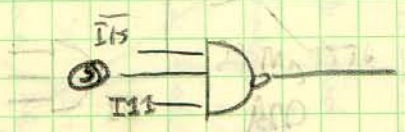




Mode D0/R7 disables Lookahead

28 May 1976 AM





1) 74S21
4) 74S11
3) 74S10
1) 74S20
2) 74S00

10 IC's

CLA 74181 SN5

The IR Decode signals drive the priority Encoder (2) 74148's and 1) 74S00) and 1 of 16 selectors to code the INSTRUCTION ROM Addresses -
The instruction ROM is a 24 x 256 array providing 12 bits of Addressing information pointing to the microcode executing the decoded instruction and 12 bits of Program flow data / Trap instructions / Byte operations / special / NOP / and look ahead coding for each instruction type.

add a conditional look ahead enable line
add a second conditional look ahead enable line
add an inhibit Trace Bit line

13 July 1976

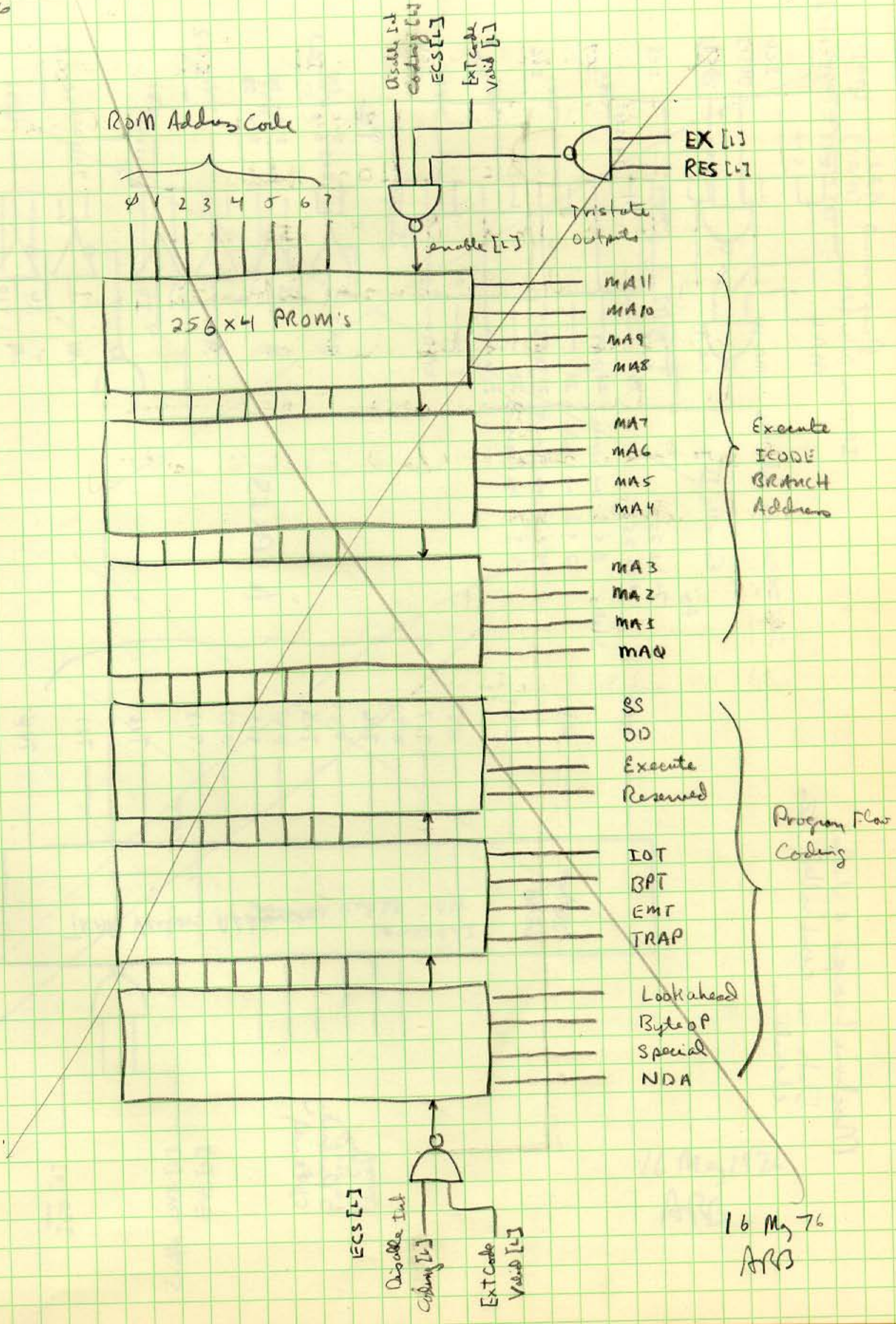
the Instruction Rom is 32 bits x necessary length
(for PDP 11 emulation length is 256)

Add PROM/RAM are accessed as memory

Note change
13 July 1976

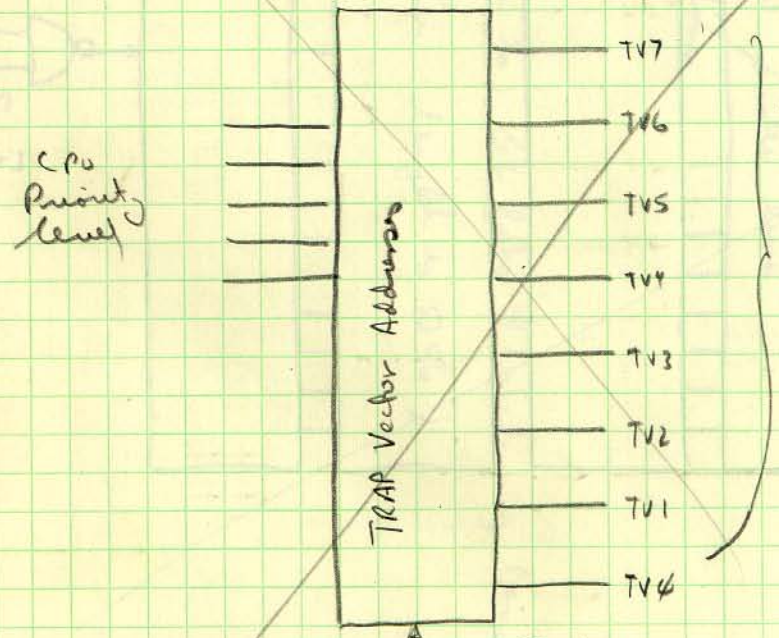
IR Decode ROM

All PROM/RAM are accessed as memory, this
Configuration is no longer valid!!!



16 May 76
ARB

Machine traps & Vector
Data is controlled ~~by~~
elsewhere



Programmed Addressing

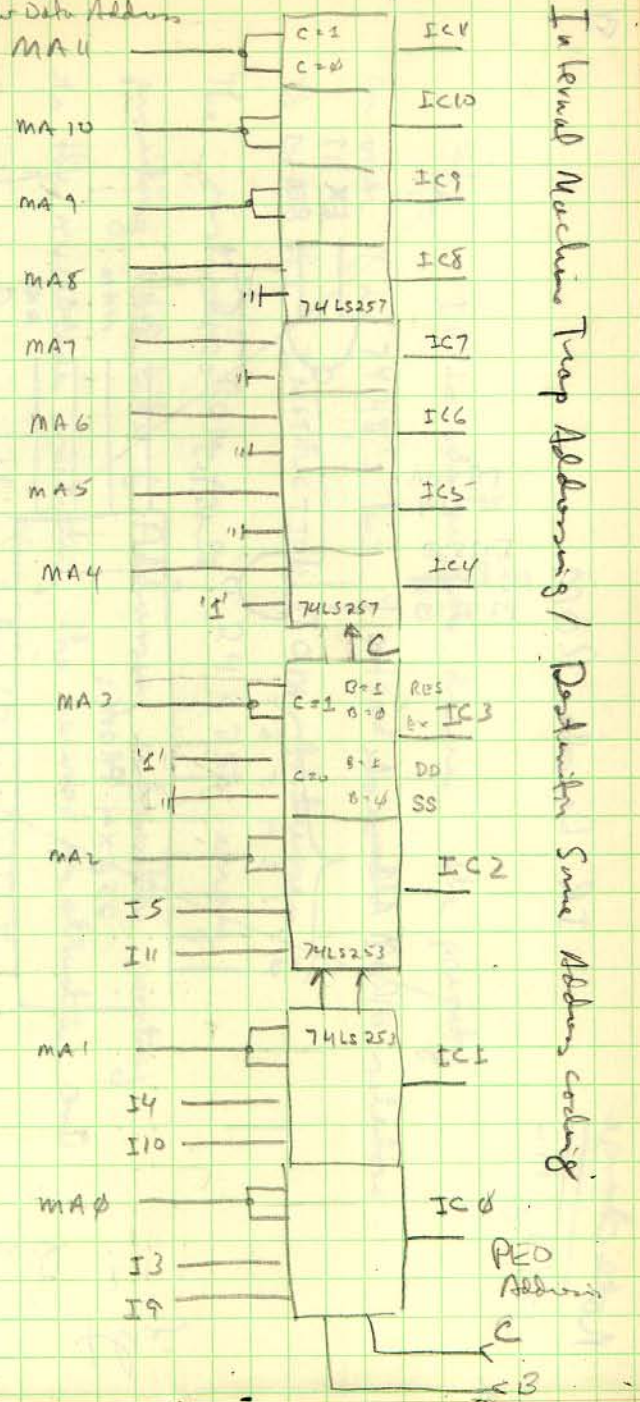
IR Flow Data Address

INT IR Decode Address

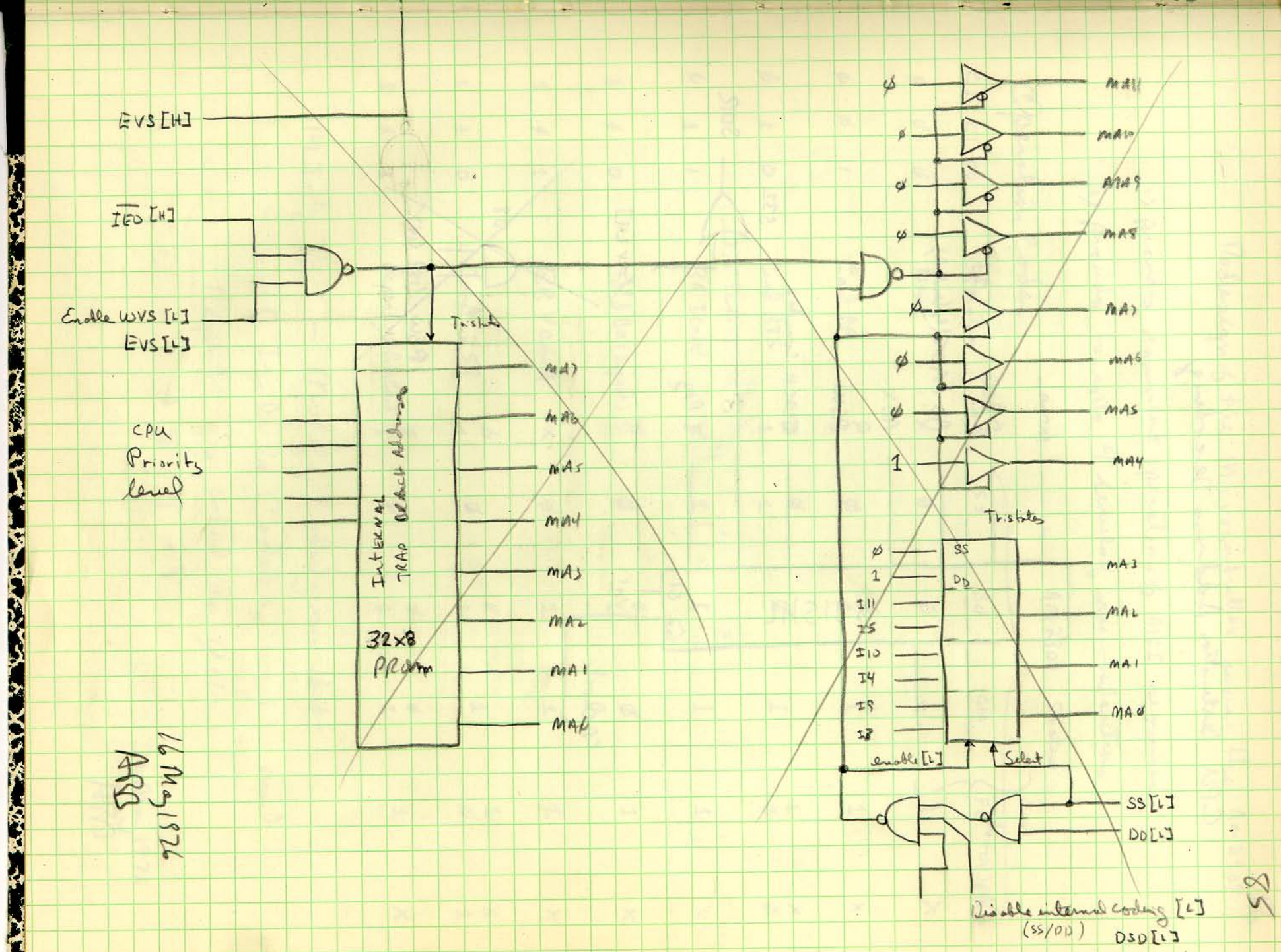
Enable [4]

to
74LS257's
74LS253's

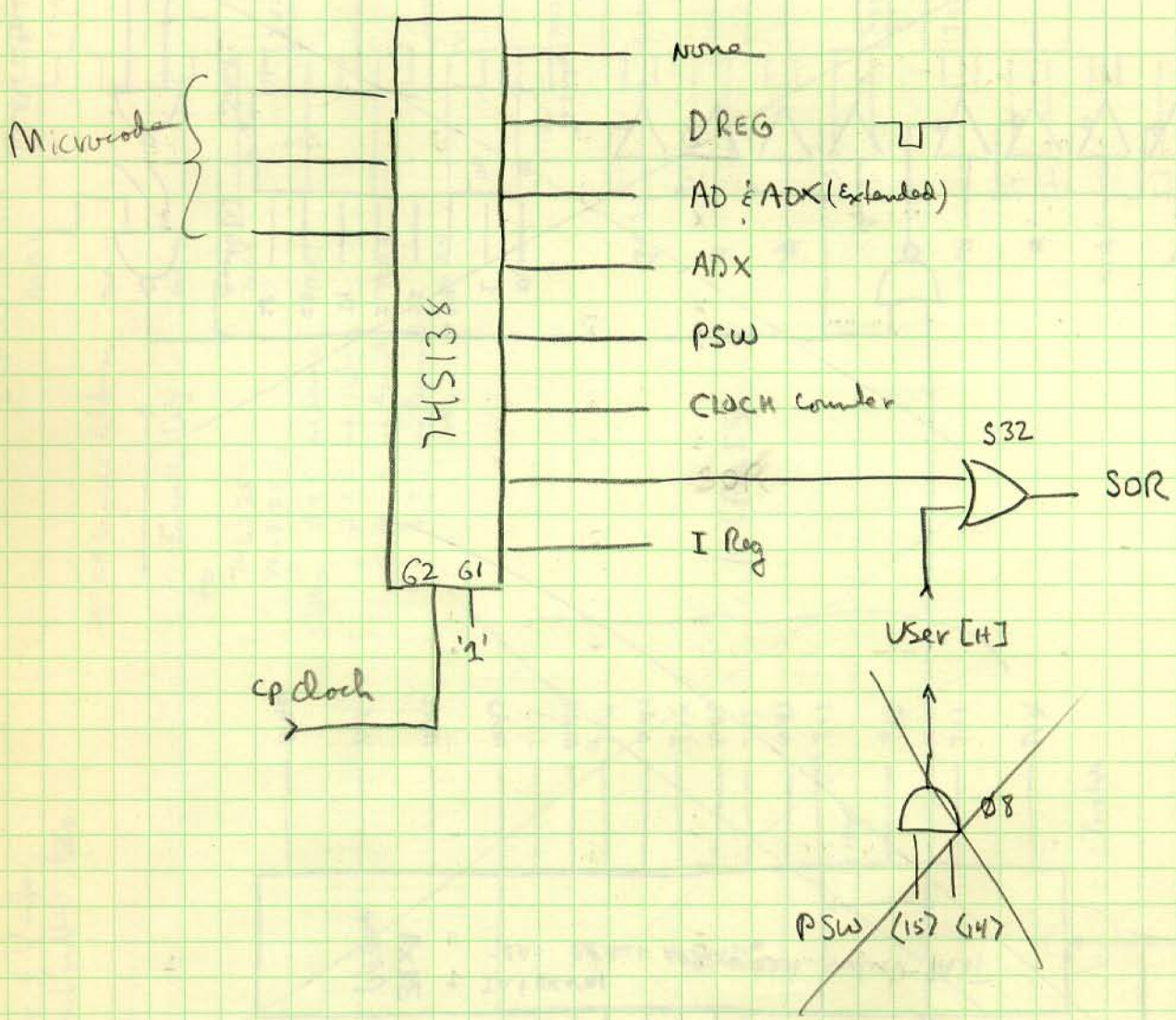
States 0 & 1 are Same
2 & 3 are Dest
4 & 5 are Exam
6 & 7 are Reserved



Interrupt Machine Trap Addressing / Dest with Same Address coding


EVS [H]
IED [H]
Enable WVS [L]
EVS [L]
CPU Priority level
Tristate
INTERNAL TRAP REACT Address
32x8 PROM
MA7
MA6
MA5
MA4
MA3
MA2
MA1
MA0
MA11
MA10
MA9
MA8
1
Tristates
SS
DD
I11
I5
I10
I4
I9
I8
enable [L]
Select
SS [L]
DD [L]
Disable internal coding [L] (SS/DD)
DSD [L]

CPU Destination Address Decoding



Redesign of the Microcontroller using the Am2909

1) provides equivalent operations of MSI sequencer
2) provides for upto 4 levels of microsubroutines

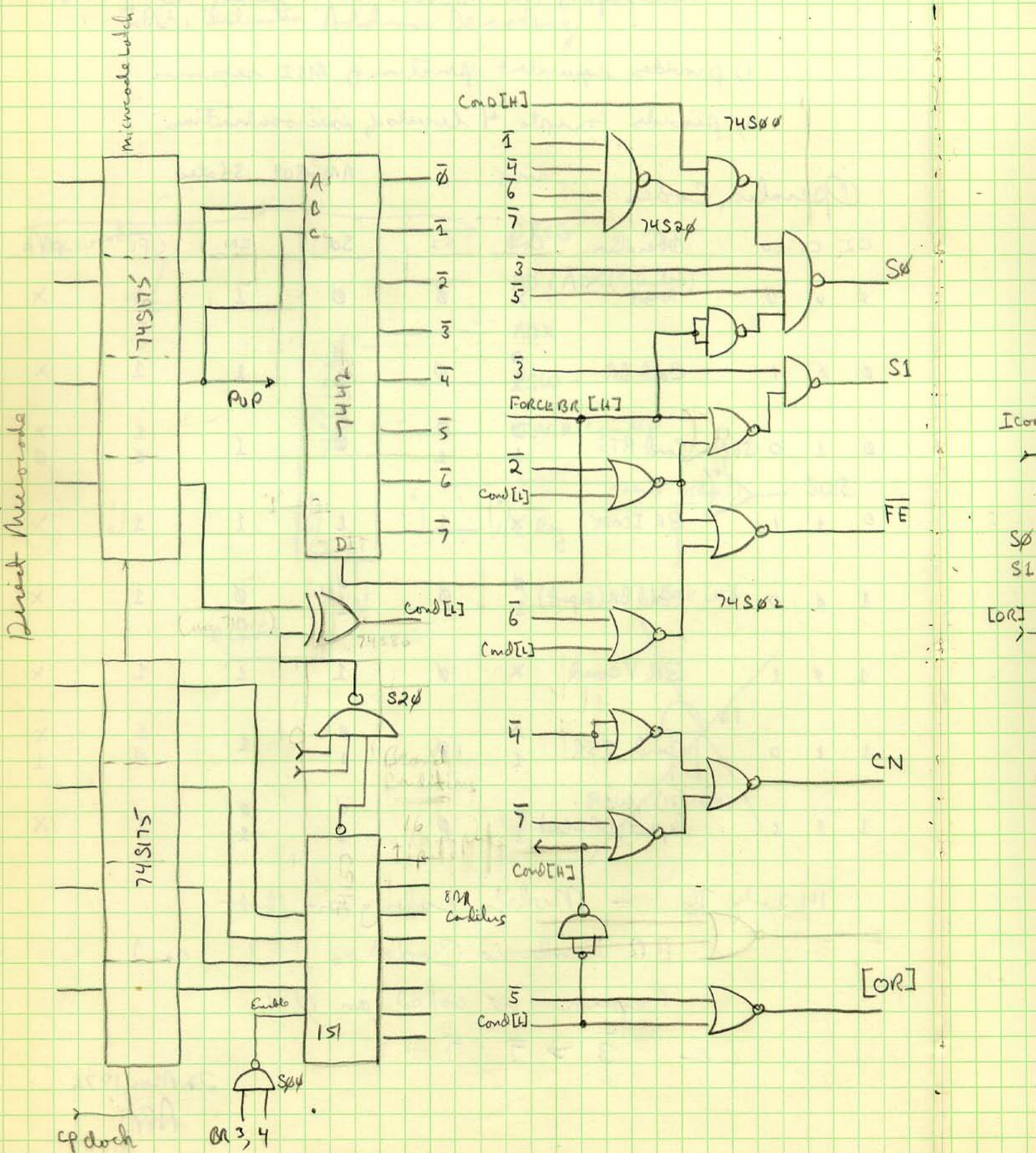
Operation Codes

C2	C1	C0	Operation	Cond	S1	S0	CN	$\overline{FE}$	PUP
					AM2909 States				
0	0	0	Next	X	0	0	1	1	X
0	0	1	Cond BR	0 1	0	0 1	1	1	X
0	1	0	Cond RTS	0 1	0 1	0	1	1 0	X 0
			(Cond JSR Icode)						
0	1	1	BR ICODE	X(0/1)	1(0/1)	1(0/1)	1	1(1/0)	X(X/1)
1	0	0	Cond BR (Repeat)	0 1	0	0 1	0	1	X
1	0	1	BR V Cond	X	0	1	1	1	X
1	1	0	Cond JSR	0 1	0	0 1	1	1 0	X 1
1	1	1	Repeat, BR on Cond	0 1	0	0 1	0 1	1	X

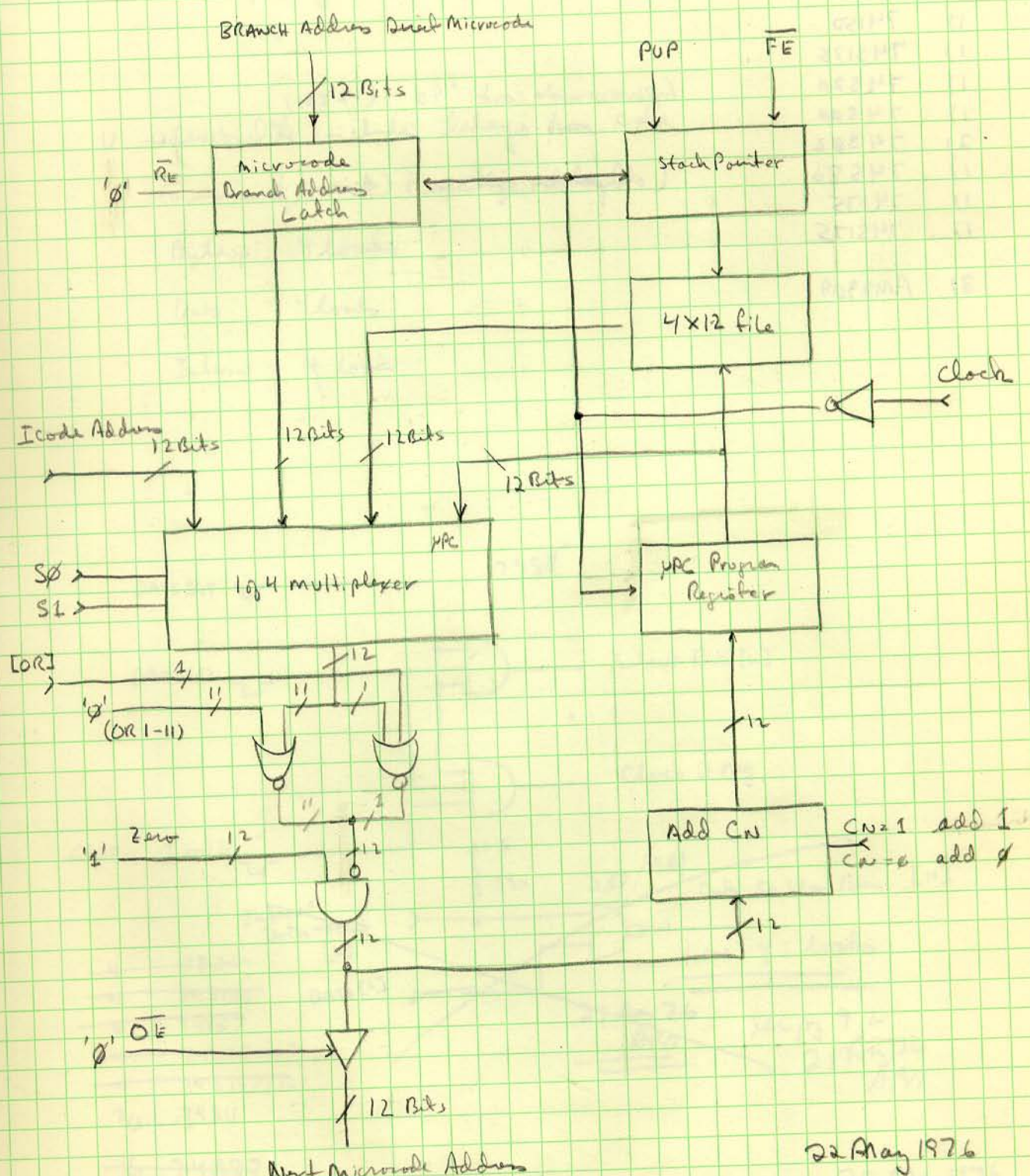
14 July 76 – Modify coding so that
BR Icode is Coded as 111 and
Repeat is coded as 011

i.e $\overline{3} \rightarrow \overline{7}$; $\overline{7} \rightarrow \overline{3}$

22 May 1976
ABV


Microcode Latch
74S175
PUP
74HC
A
B
C
D
Ø
1
2
3
4
5
6
7
Cond[H]
74SØØ
74S2Ø
SØ
S1
FORCEBR [H]
Cond[L]
FE
74SØ2
Cond[L]
S2Ø
74S175
CN
Cond[H]
BR Address
[OR]
Enable
151
SØ4
cp clock
BR 3, 4
Decent Microcode

3 AM2909 units connected for 12 Bit controller



22 May 1976

ARD

IC's used

1) 74150
1) 74S138
1) 74S20 Approximate cost $90 (12 ICs)
1) 74S00
2) 74S02
1) 74S86 (old version = $70 (31 ics))
1) 74175
1) 74S175

3) AM2909

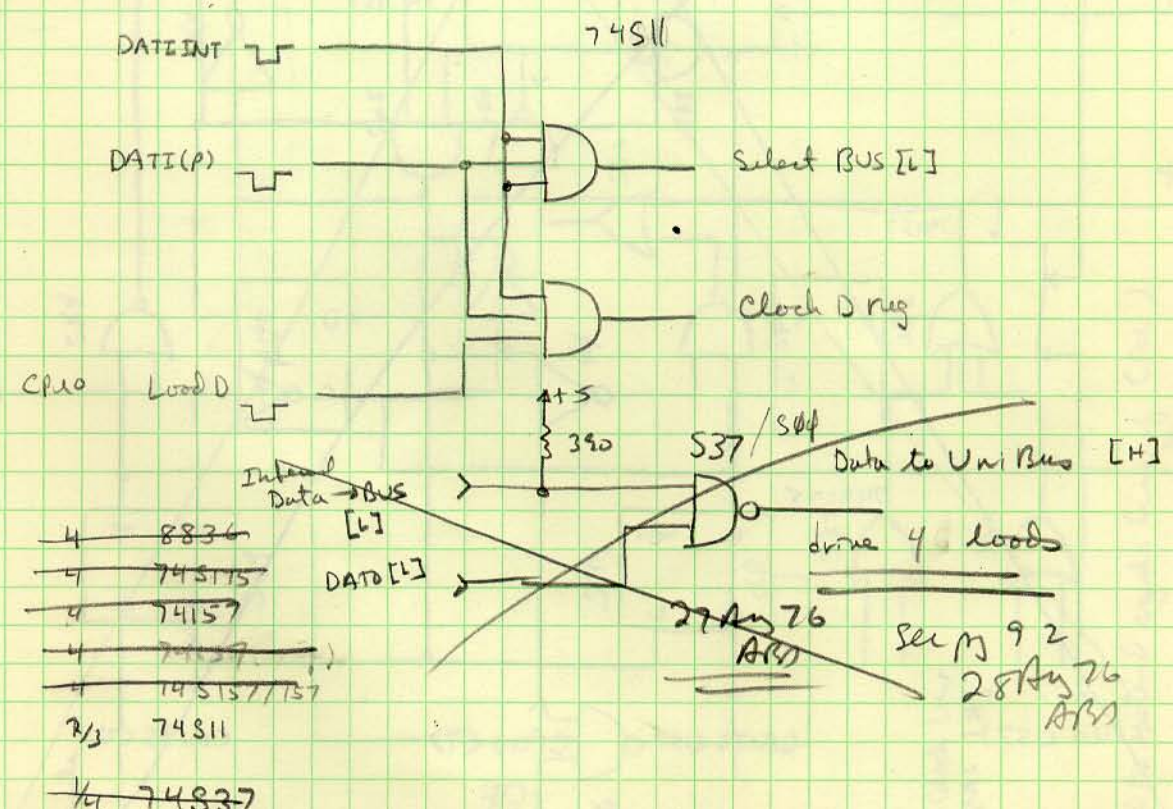
Redesign of Data Register / BUS interface

1) redesigned to include delays from CPU
to minimize cost & equalize delays

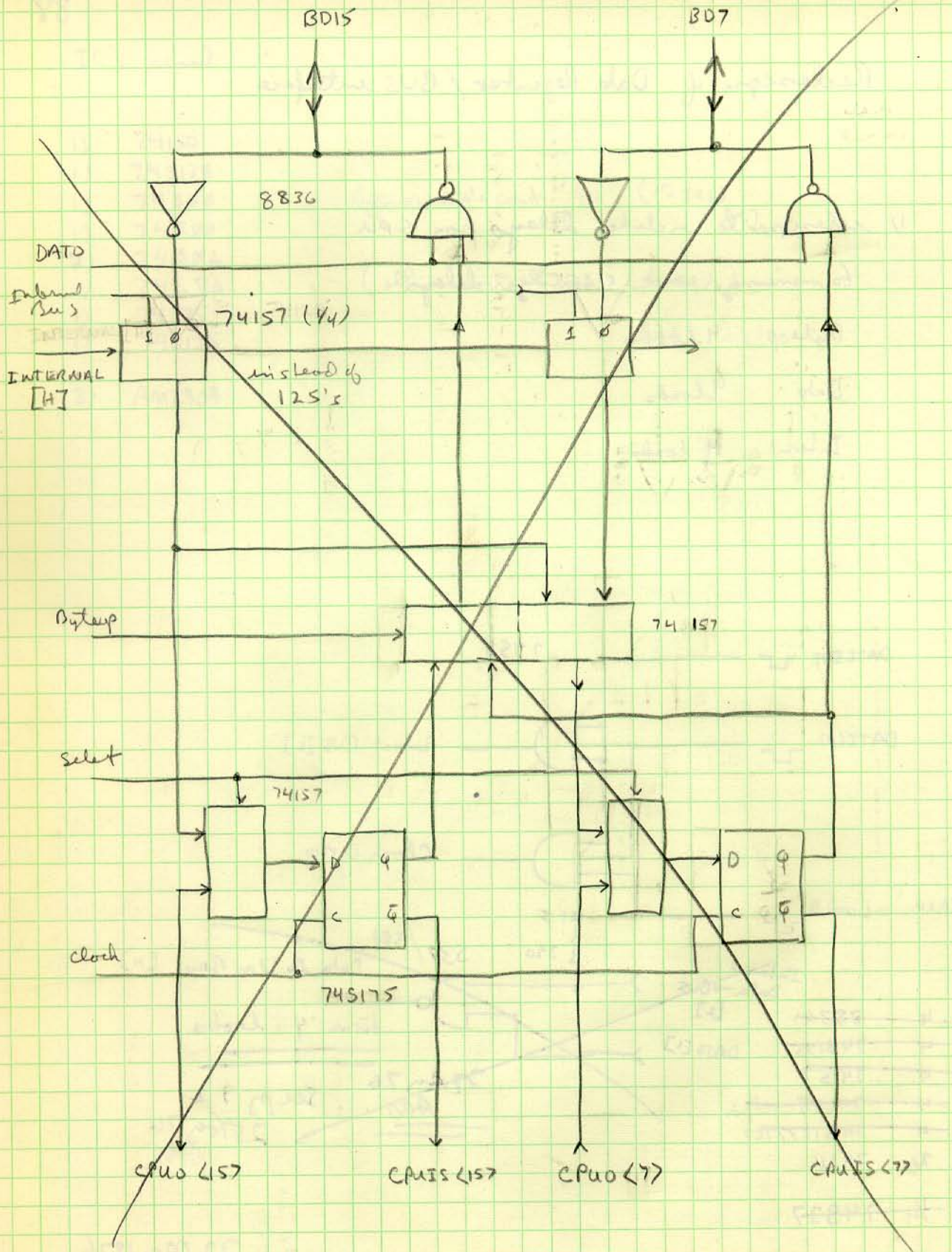
Bytesp 4 loads

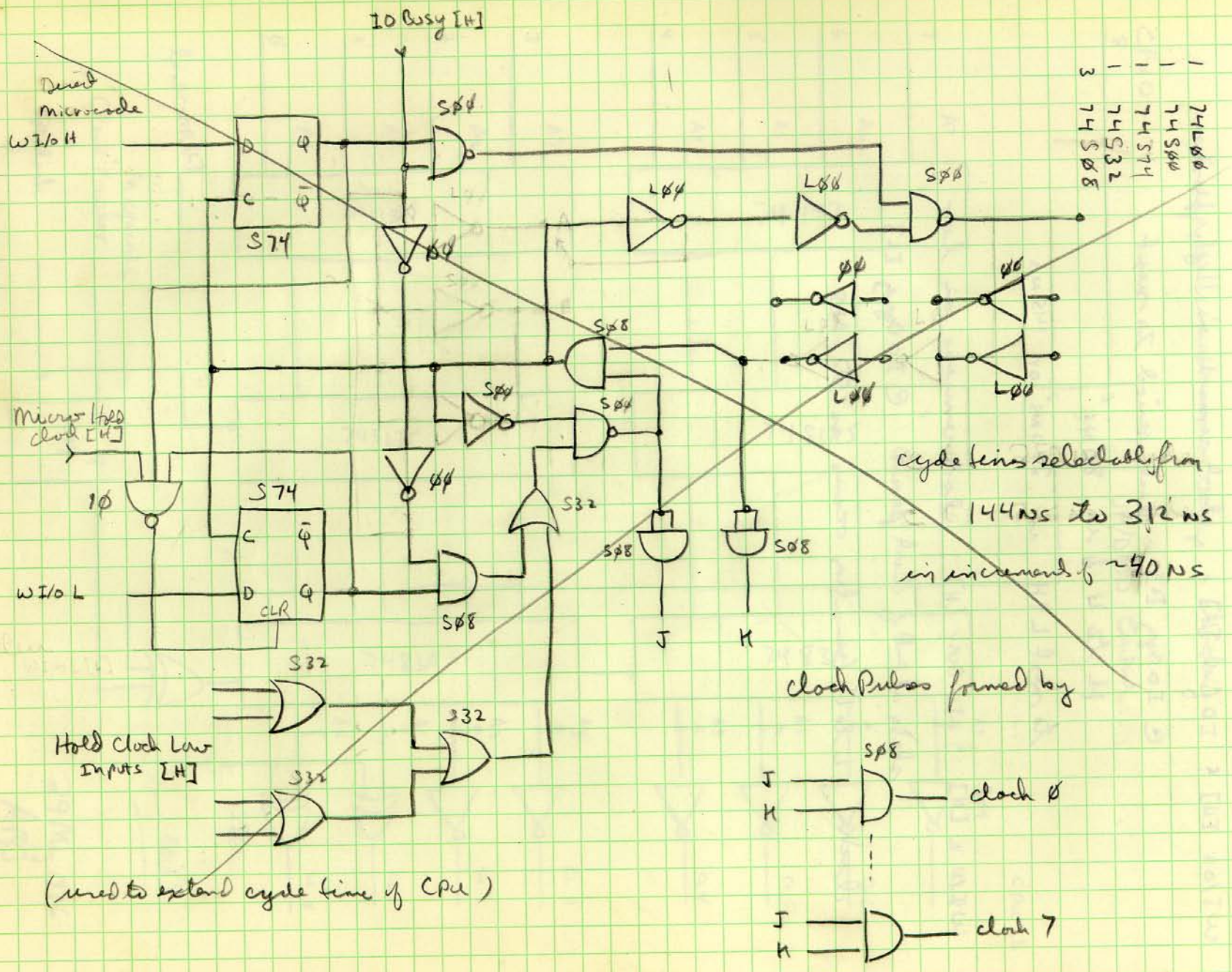
Data 4 loads

Interval 4 loads



22 May 1976
ARS


BD15
BD7
8836
DATO
External Bus
74157 (1/4)
I 0
INTERNAL [H]
misload of 125's
Bytesp
74 157
select
74157
D Q
C Q
clock
74S175
CPUD<15>
CPUIS<15>
CPUD<7>
CPUIS<7>
IO Busy [H]


IO Busy [H]
Decod Microcode
WI/O H
S74
SØØ
LØØ
LØØ
SØØ
ØØ
ØØ
ØØ
ØØ
SØ8
SØØ
SØØ
Micro Hold clock [H]
1Ø
S74
S32
WI/O L
CLR
SØ8
SØ8
SØ8
J
H
S32
S32
S32
Hold Clock Low Inputs [H]
(used to extend cycle time of CPU)
1 74L00
1 74S00
1 74S74
1 74S32
3 74S08
redesigned CPU Clock
cycle time selectable from
144NS to 312 NS
in increments of ~40 NS
clock Pulses formed by
SØ8
J
H
clock Ø
J
H
clock 7
89

WI/OH [H] & IO BUSY [H] clock remains in High position

@ IO BUSY [L] clock period to next

H to L is 80 NS

to next L↑H is 220 NS or 145 NS

WI/O L [H] & IO BUSY [H] clock remains in low

state at end of clock period @ IO BUSY [L]

clock L to H transition occurs in 50 NS

CP

8

7

6

5

4

3

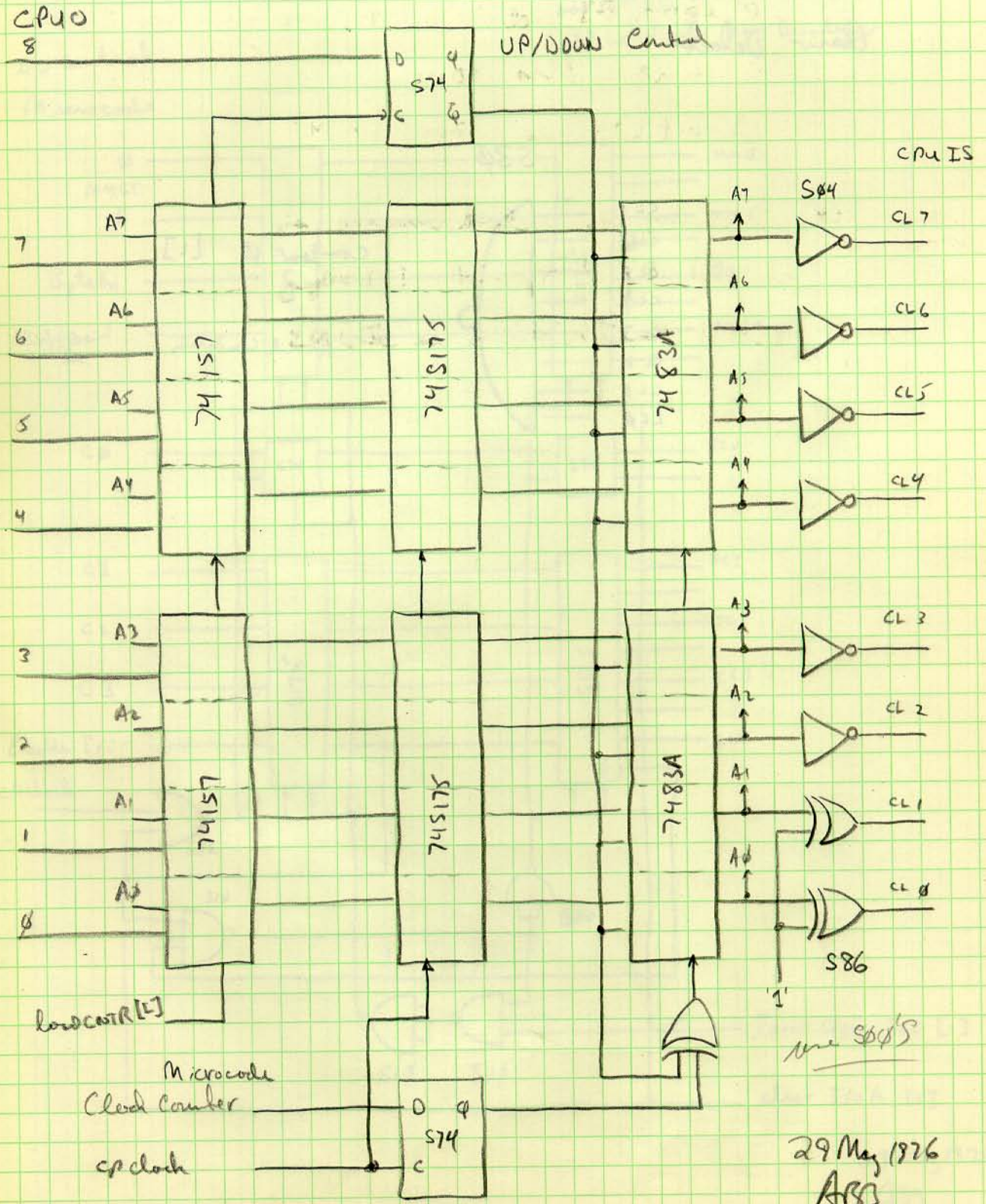
2

1

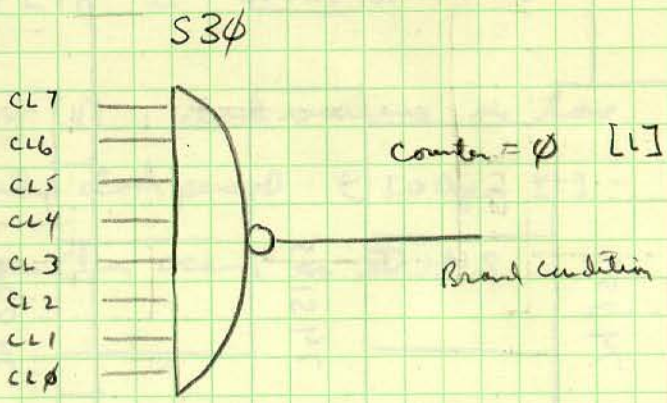
0

Redesign of Counter logic

a) allows simultaneous setup & clocking



Zero Detect

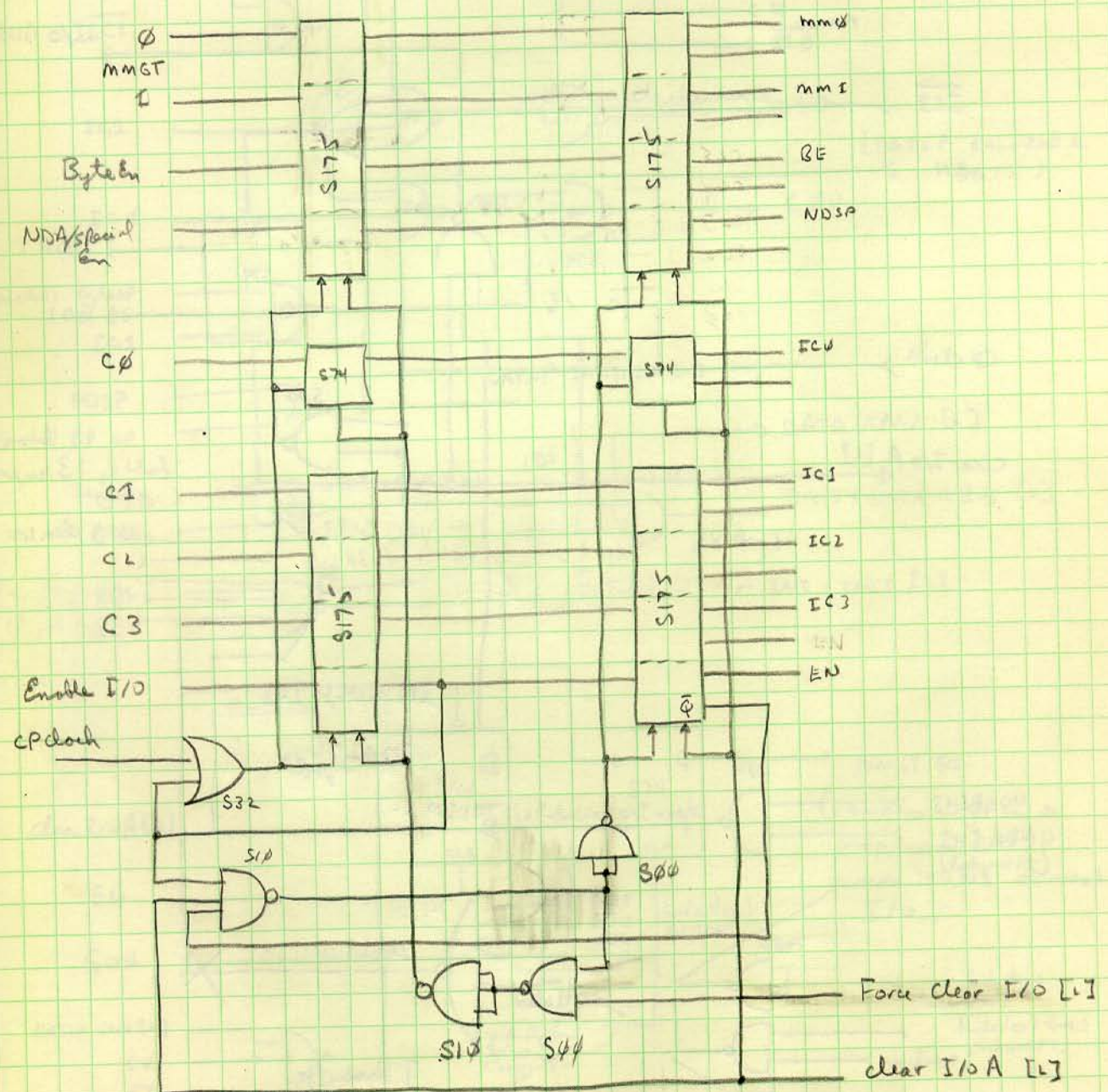

S3ϕ
CL7
CL6
CL5
CL4
CL3
CL2
CL1
CLϕ
counter = ϕ [L]
Branch condition

Redesigned I/O Control Logic

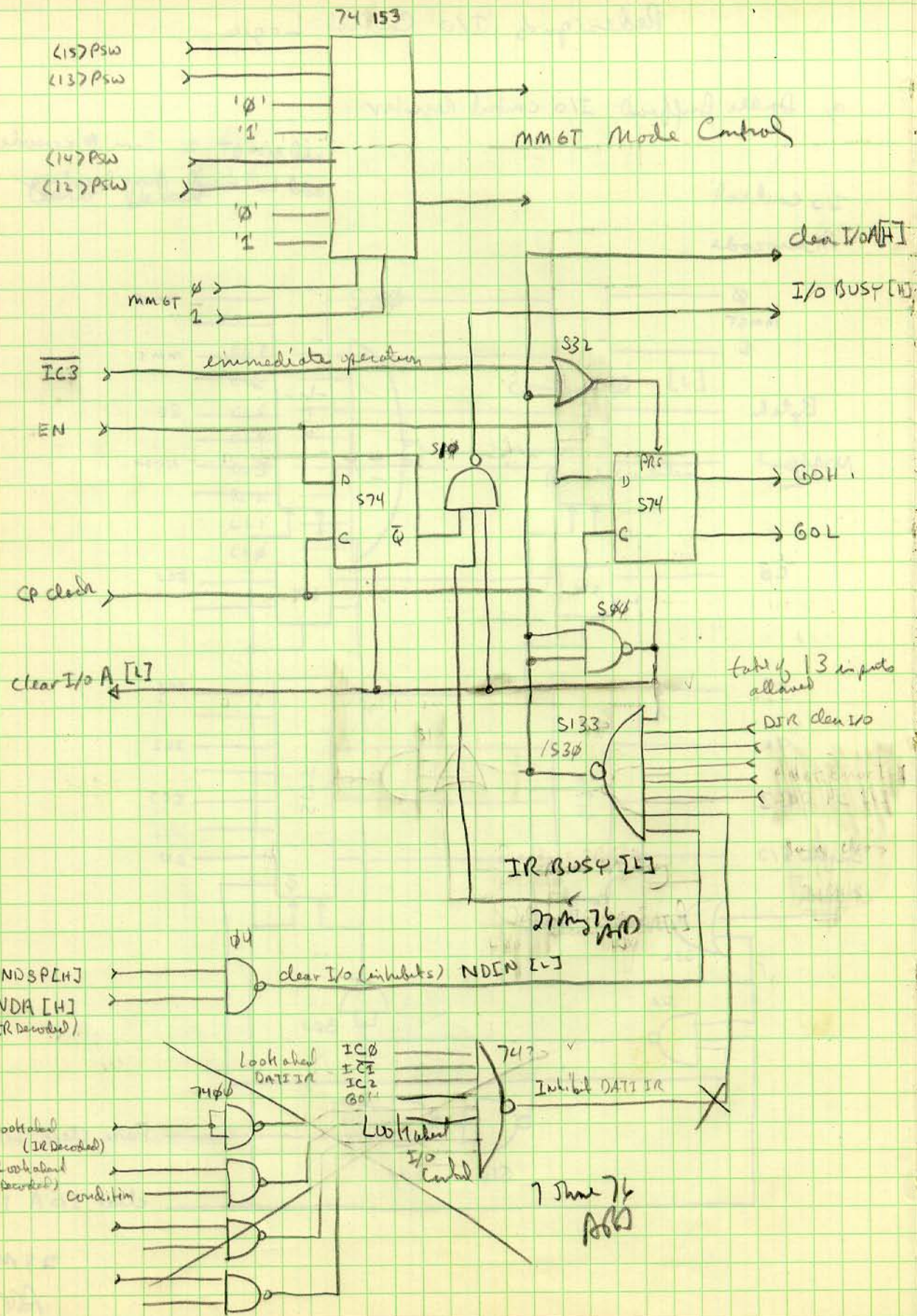
a) Double Buffered I/O control Register

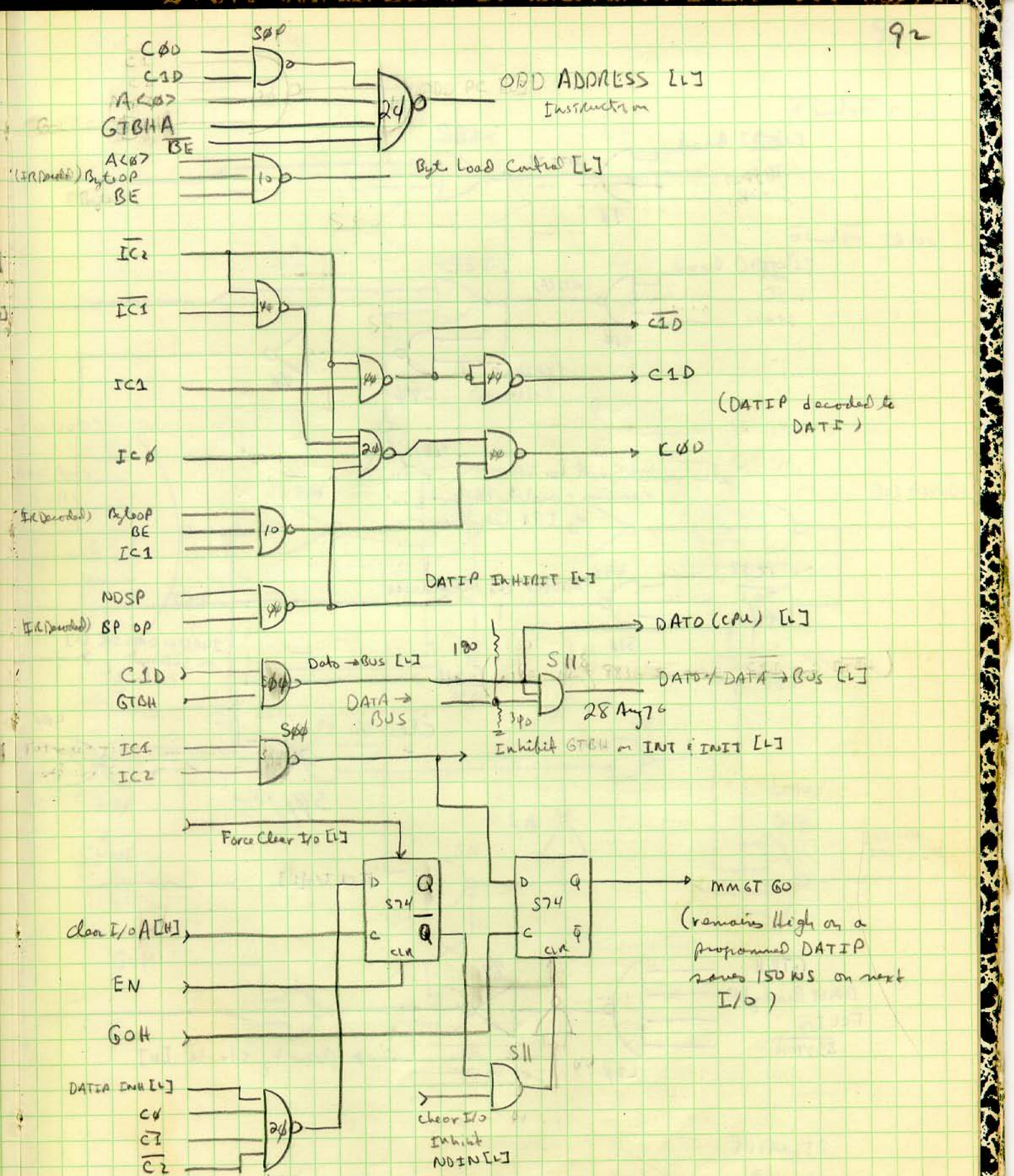
add mmGT 2 from Microcode
add Lockout from Decoding

I/O control Microcode

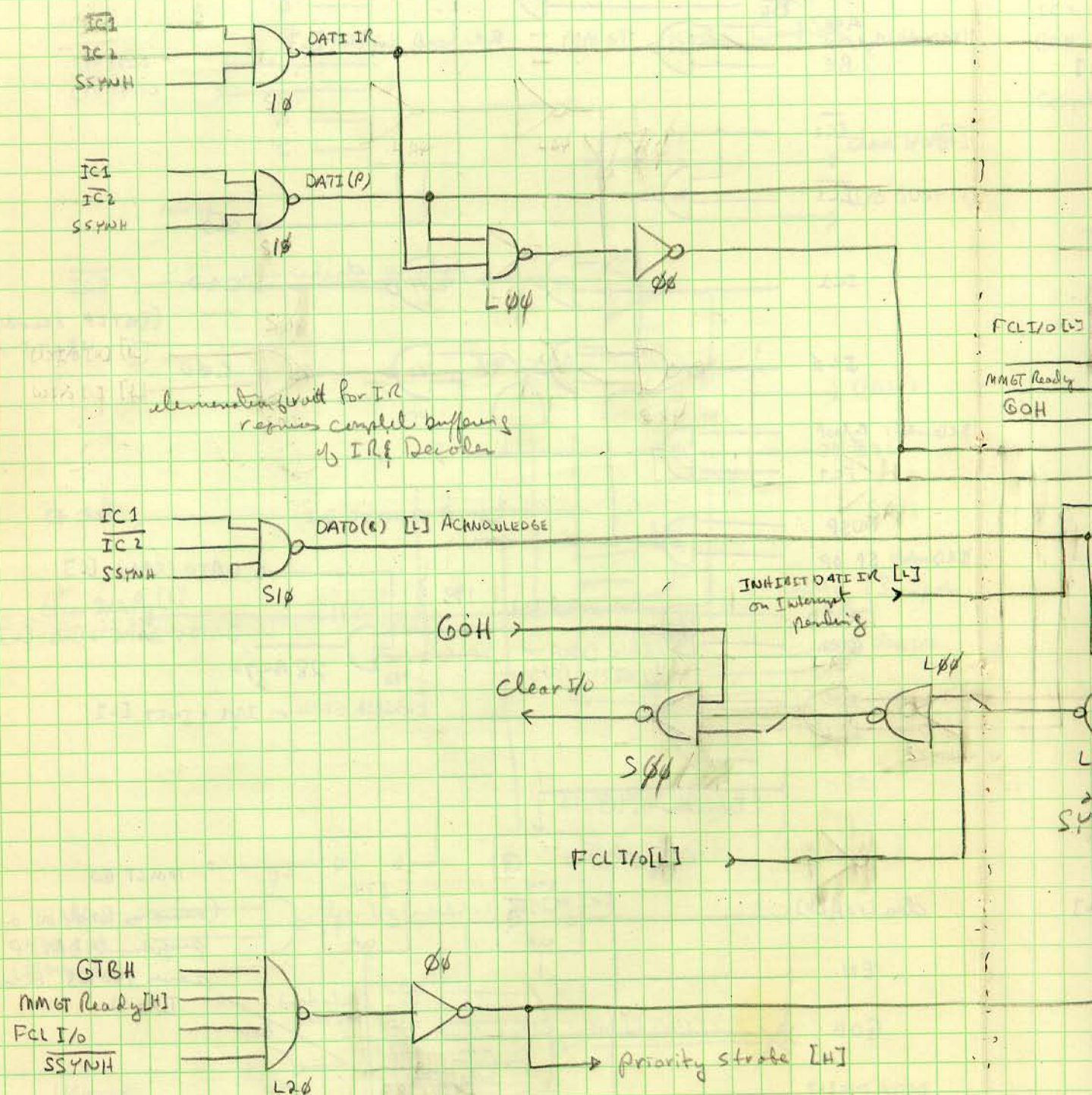


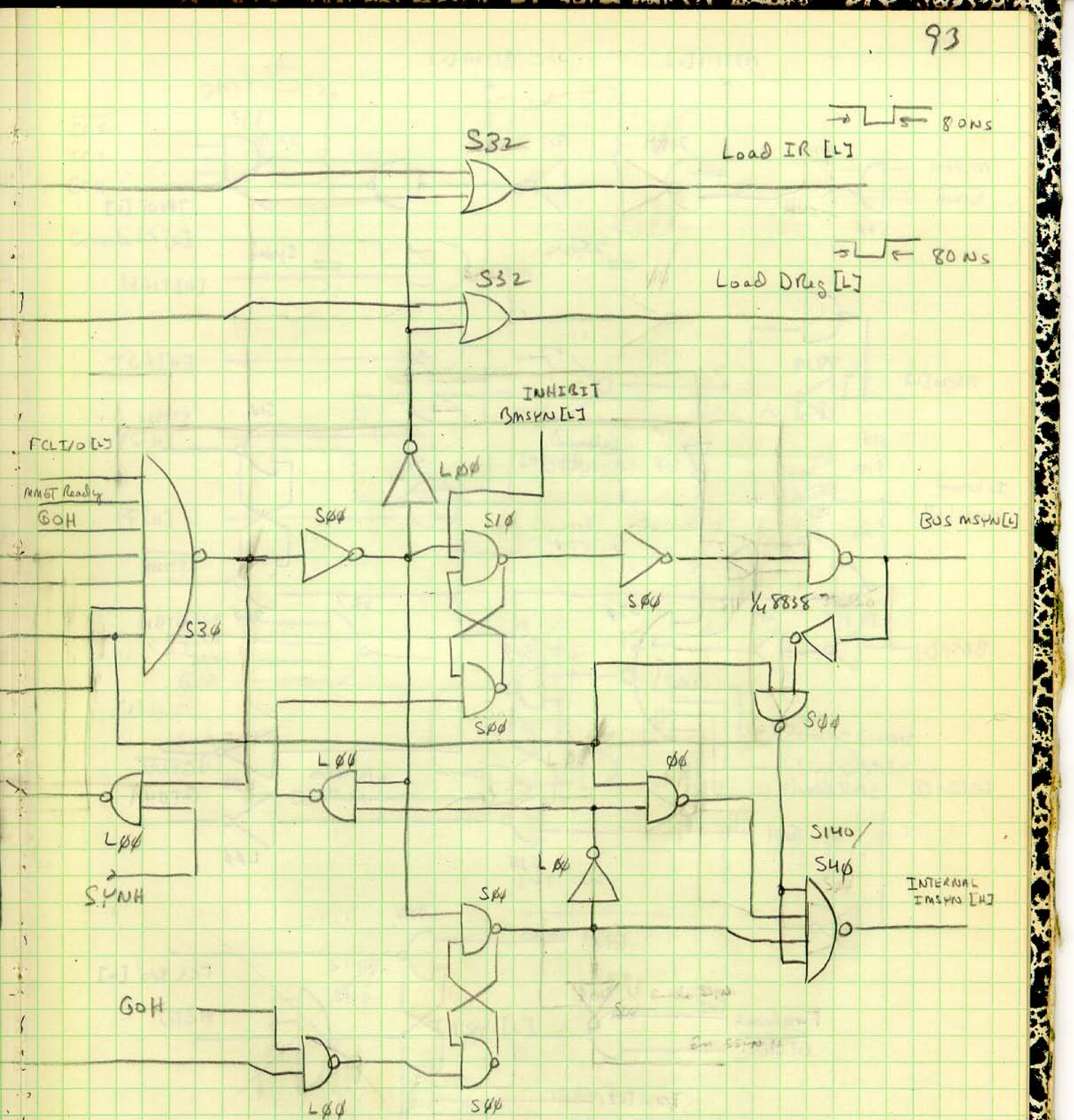
29 May 1976
ARM


74 153
<15> PSW
<13> PSW
'0'
'1'
<14> PSW
<12> PSW
'0'
'1'
MM 6T Mode Control
clear I/O A [H]
I/O BUSY [H]
mm 6T
0
1
IC3
immediate operation
S32
EN
S10
D
S74
C
Q
PRS
D
S74
C
GOH
GOL
CP clock
S00
clear I/O A [L]
fan-in 13 inputs allowed
S133
/S30
DIR clear I/O
IR BUSY [L]
27 May 76 ABD
04
NDSP [H]
NDIR [H]
(IR Decoded)
clear I/O (inhibits) NDIN [L]
Lookahead DATI IR
74 00
IC0
IC1
IC2
BOH
Lookahead I/O cond
74 13
Inhibit DATI IR
Lookahead (IR Decoded)
Cond Lookahead (IR Decoded)
Condition
7 June 76 ABD

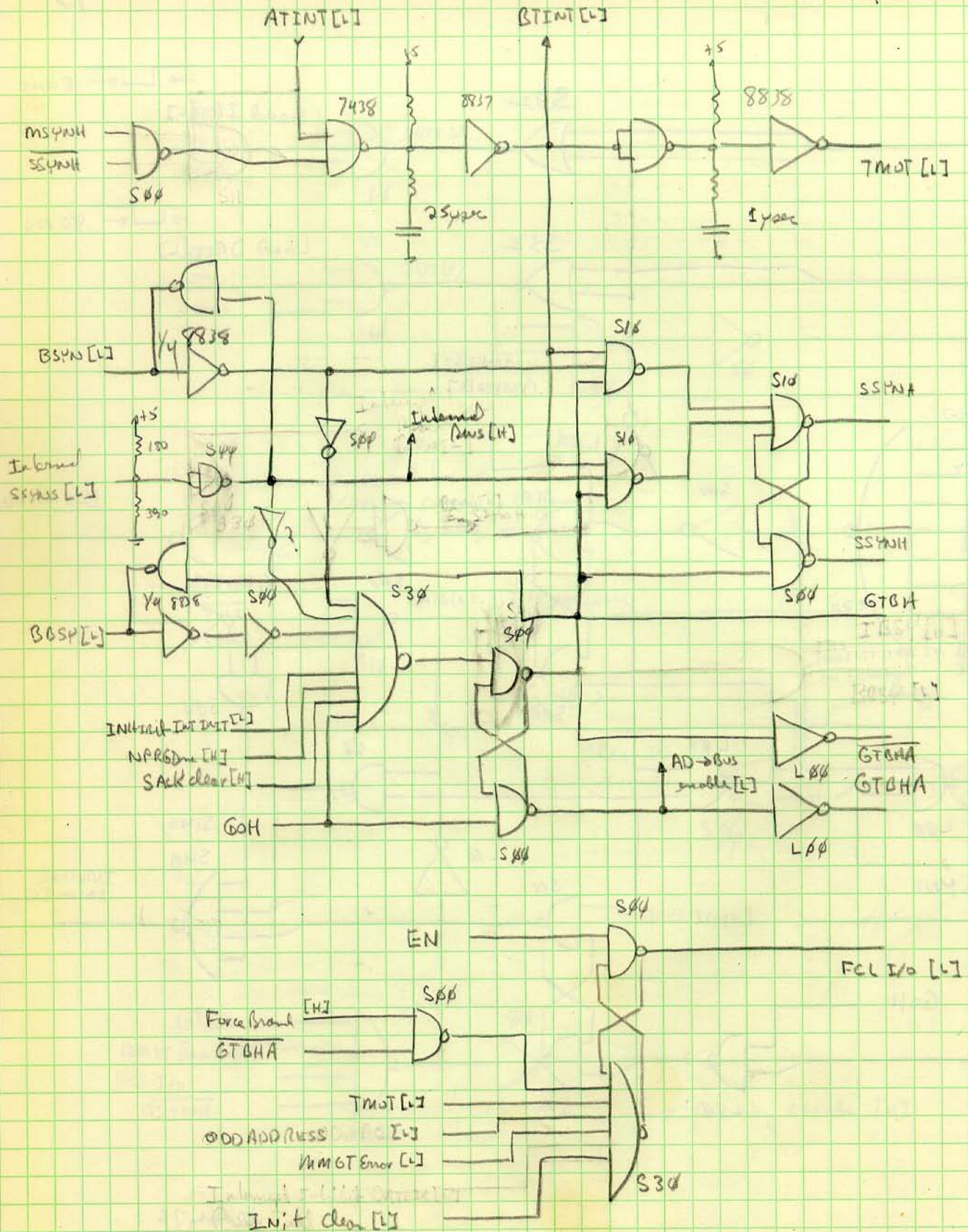


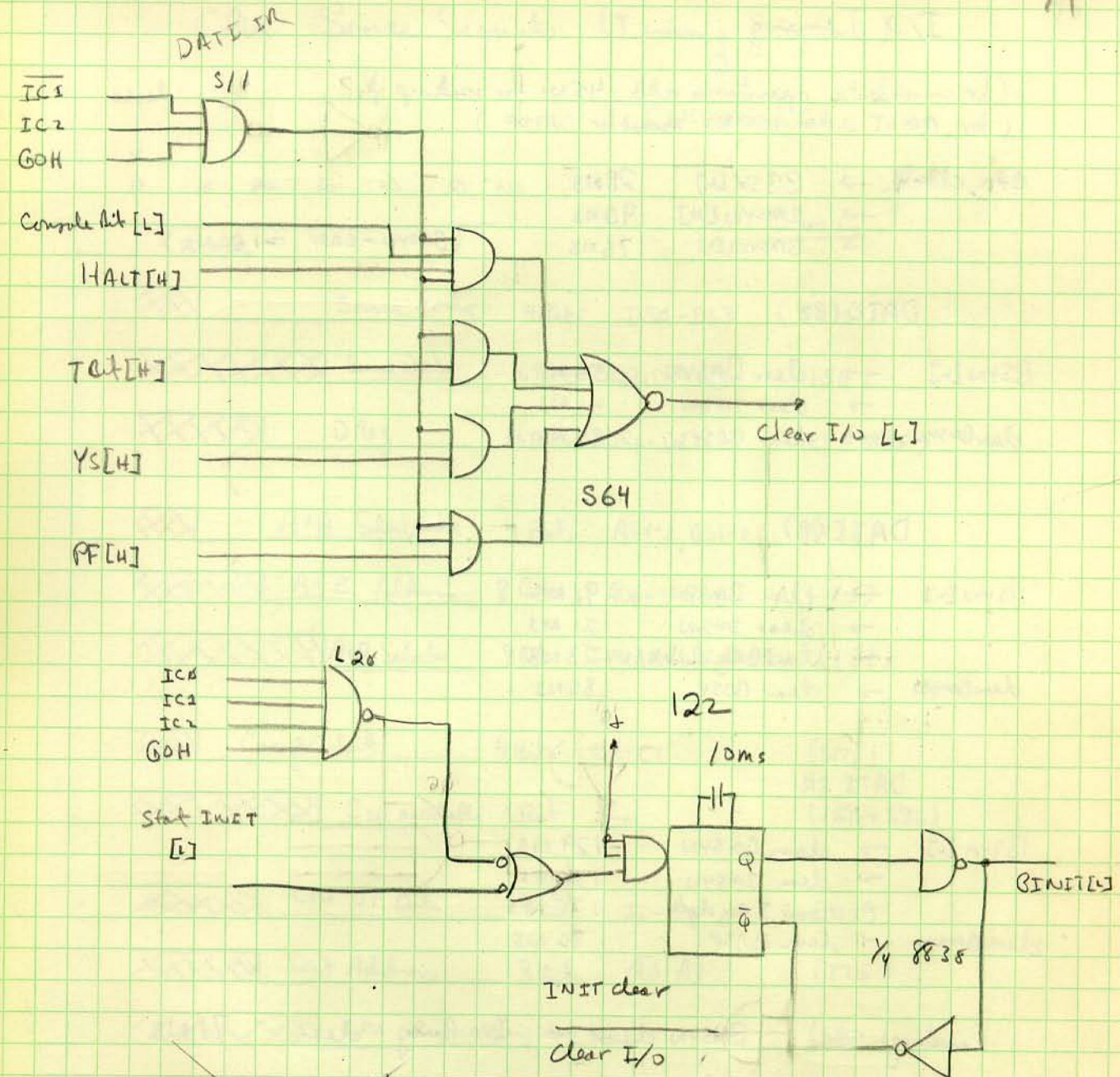
29 May 1976
ARB


IC1
IC2
SSYNH
1ø
DATI IR
DATI(P)
Løø
øø
FCLI/O [L]
MMGT Ready
GOH
elimination of wait for IR
requires complete buffering
of IR & Decoder
IC1
IC2
SSYNH
S1ø
DATO(B) [L] ACKNOWLEDGE
INHIBIT DATI IR [L]
on Interrupt
pending
GOH
Clear I/O
Søø
Løø
FCLI/O[L]
GTBH
MMGT Ready[H]
FCL I/O
SSYNH
L2ø
øø
Priority strobe [H]


80 NS
S32
Load IR [L]
80 NS
S32
Load DReg [L]
INHIBIT
BMSYN [L]
FCLI/O [L]
MMGT Ready
GOH
L ØØ
SØØ
S1Ø
BUS MSYN [L]
S3Ø
SØØ
¼ 8838
SØØ
SØØ
L ØØ
ØØ
L ØØ
SYNH
S1HO/
S4Ø
L ØØ
SØØ
INTERNAL IMSYN [H]
GOH
L ØØ
SØØ

30 May 76
ARD


ATINT[L]
BTINT[L]
+5
7438
8837
8838
MSYNH
SSYNH
SØØ
2.5μsec
1μsec
TMOT[L]
BSYN[L]
Y4 8838
S1Ø
S1Ø
SSYNA
Internal SSYNS[L]
180
390
SØØ
Internal BUS[H]
SSYNH
BBSY[L]
S3Ø
GTBH
INHIBIT INTRPT[L]
NPRGRnt[H]
SACK clear[H]
GOH
AD→BUS enable[L]
LØØ
GTBHA
GTBHA
LØØ
EN
FCL I/O[L]
Force Branch
GTBHA
[H]
TMOT[L]
ODD ADDRESS [L]
NXM GT Error[L]
Init clear[L]



Not necessary 28 Aug 76
ARB
replaced with ADD/Data Multiplexers

31 May 1976
ARB

I/O Timing

(for immediate operations add 45 NS for setup)
(for MM6T add 150 NS except for DATIP)

CPU clock	→ BBSY [L]	28 NS	
	→ IMSYN [H]	90 NS	
	→ BMSYN [L]	176 NS	(BMSYN − BBSY = 150 NS)

DATO(B)

BSYN [L]	→ clear BMSYN	56 NS	
	→ clear IMSYN	41 NS	
clear BMSYN	→ clear BBSY	80 NS	

DATI(P)

BSYN [L]	→ clear BMSYN	96 NS	
	→ clear IMSYN	121 NS	
	→ load DReg clock	123 NS	
clear BMSYN	→ clear BBSY	80 NS	

DATI IR

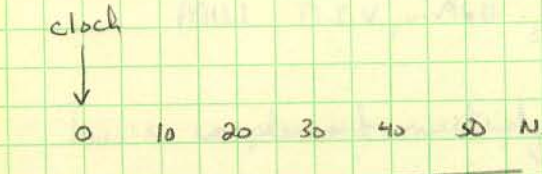
BSYN [L]	→ clear BMSYN	129 NS	
	→ clear IMSYN	154 NS	
	→ load IReg clock	123 NS	
clear BMSYN	→ clear BBSY	80 NS	

timing for BMSYN clear → I/O Busy clear 71 NS

cc I/O Busy clear → clear BBSY 9 NS

timing need to be recomputed after changes 17 Jul 76

CPU Source Signals (Timing Diagrams)



			Devices
Source Code	4 Bits	DS0-DS3	(S175)
Source Data	16 Bits	CPUD0-D15	(150)
D Reg	16 Bits	DR0-DR15	(S138 + S175)
A & A Select Code	4 Bits	AS1-2, BS1-2	(S175)
A & B Address	8 Bits	AD0-3, BD0-3	(175)
A & B Selector	8 Bits	CPUA0-3, CPUB0-3	(153)
Carry IN Select	4 Bits	C0-C3	(S175)
C_{IN} available	1 Bit	C_N	(151 + S86)
CPU OP Data	9 Bits	I_0-I_8	(175)
Dest Address	3 Bits	A0-A3	(175)
CPU OP Decoding	9 Bits		(S157 + others)
C, Z, V, N Control	16 Bits		(175)
Shift-Rotate	8 Bits	SR0-SR7	(175)
I/O BUS Control	9 Bits		(S32 + S175)
Branch Cond Decode	4 Bits		(S175)
Controller Code	4 Bits		(S175)

31 May 1976
AHO

Extended Instruction Set (EIS) to include

MUL, DIV, ASH, ASHC

Two's complement multiply, assumes source data is in DI

Execution time is 6.5 μsecs exclusive of IR + Sometimes

① H/S → CNTR

② ϕ → T1

③ WI/O [H], DI → shift LSB → Tϕ; LO/RI → C

④ Tϕ → Q

⑤ [MUL] CP; C=1; RSX + T1 → T1
C=ϕ; ϕ + T1 → T1; shift LSB BR [MUL] on $\overline{\text{CNTR}}$
T1<ϕ> → Q<15>; Q LO/RI → C; CPUV ∀ CPUN → T1<15>

⑥ C=1 T1 − RSX → T1; shift LSB
C=ϕ T1 − ϕ → T1
T1<ϕ> → Q<15>; CPUV ∀ CPUN → T1<15>

high order in T1, Low order in Q

⑦ T1 → RSX; V = CPUZ (V=1 if T1 = ϕ)

⑧ $\overline{\text{T1}}$; C = CPUZ (C=1 if T1 = −1)

⑨ Q → RSXV1; N = CPUN; Z = CPUZ; BR [J1] on C = 1

⑩ R7 → AD; BR [J2] on V=1; DATI IR

⑪ T1; N = CPUN; C = ϕ; V = ϕ; Z = ϕ; BR [BEGINX]

⑫ [J1] V = ϕ, Z = ϕ; BR [K]VN; R7 → AD; DATI IR

⑬ [J2] V = ϕ, N = ϕ; BR [K]V$\overline{\text{N}}$

⑭ [K]Vϕ C = 1; BR [BEGINX]

⑮ [K]V1 C = ϕ; BR [BEGINX]

1 June 1976

Redesigning the High Order Priority Microword

Priority are defined as

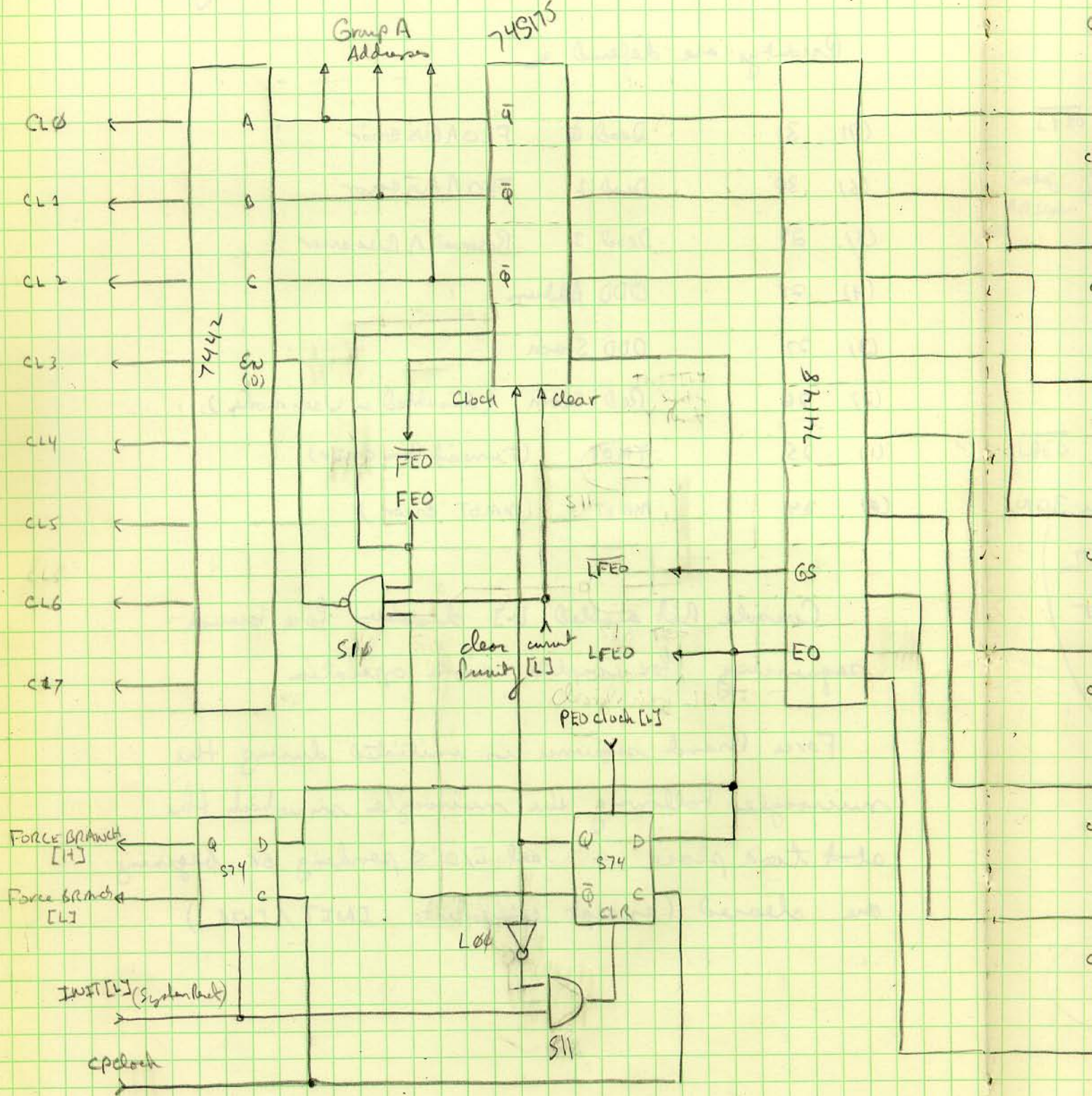
(7)	31	Dead Q	FEO ∧ Bus Error
(6)	30	Dead 1	TEO ∧ Bus Error
(5)	29	Dead 3	Reserved ∧ Bus error
(4)	28	ODD Address	
(3)	27	ODD Stack	
(2)	26	Red Stack	(disabled in User mode)
(1)	25	TMOT	(Timeout Bus Error)
(0)	24	MMGT	(MMGT Error)

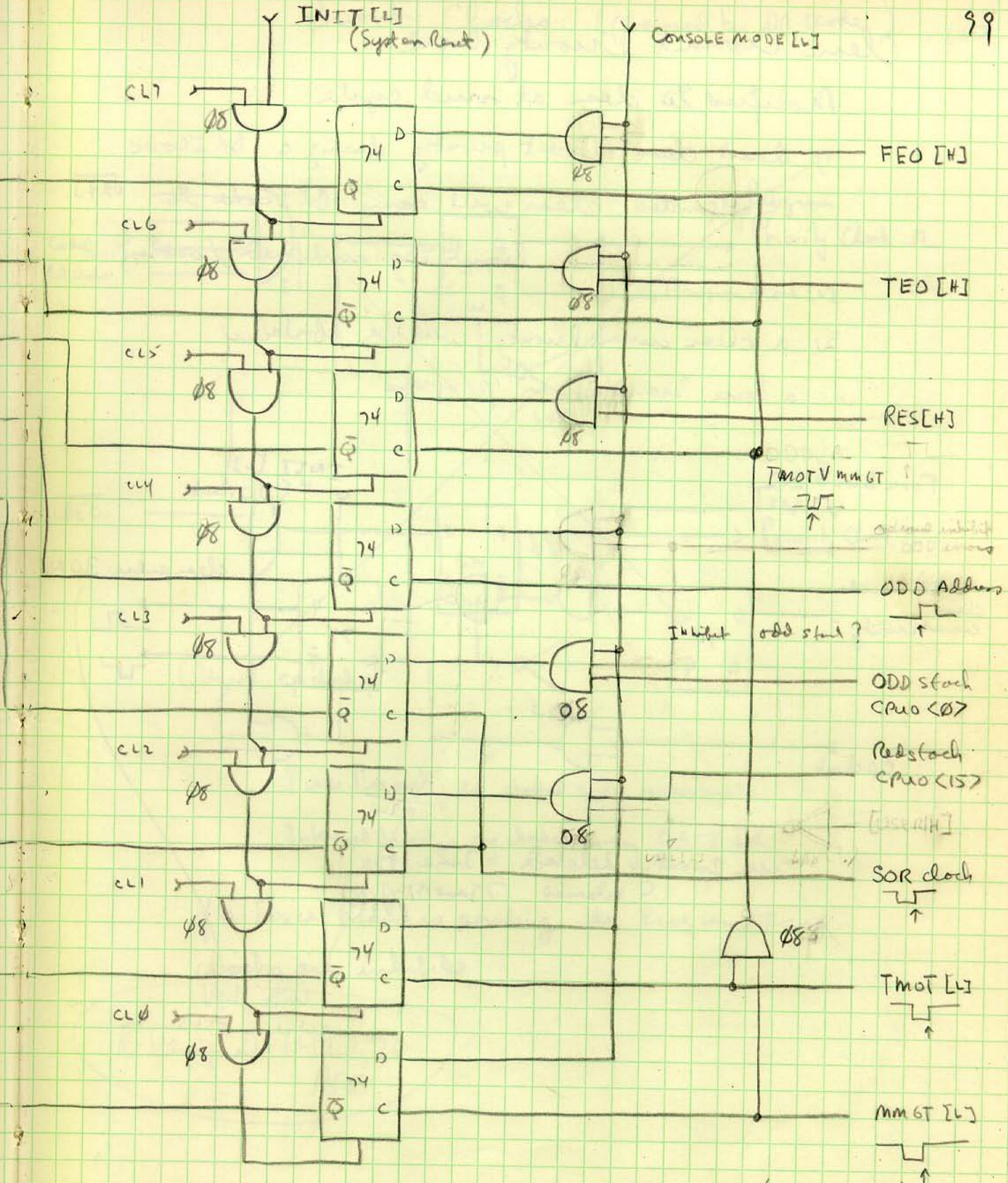
Console Bit enabled [L] disables force branch sequencing for complete console operation

Force Branch sequence is initiated during the microcycle following the microcycle in which the abort took place - all I/O's pending or beginning are cleared (except immediate INIT/INTR)

6 June 1976

BBN


Group A Addresses
74S175
CL0
CL1
CL2
CL3
CL4
CL5
CL6
CL7
A
B
C
EN (0)
7442
Q
Q̄
Clock
clear
FEO
FEO
S11
LFEO
GS
LFEO
EO
S10
clear current Density [L]
PEO clock [L]
FORCE BRANCH [H]
Force Branch [L]
S74
D
C
CLR
L0φ
INIT [L] (System Reset)
S11
cpclock



6 June 1976

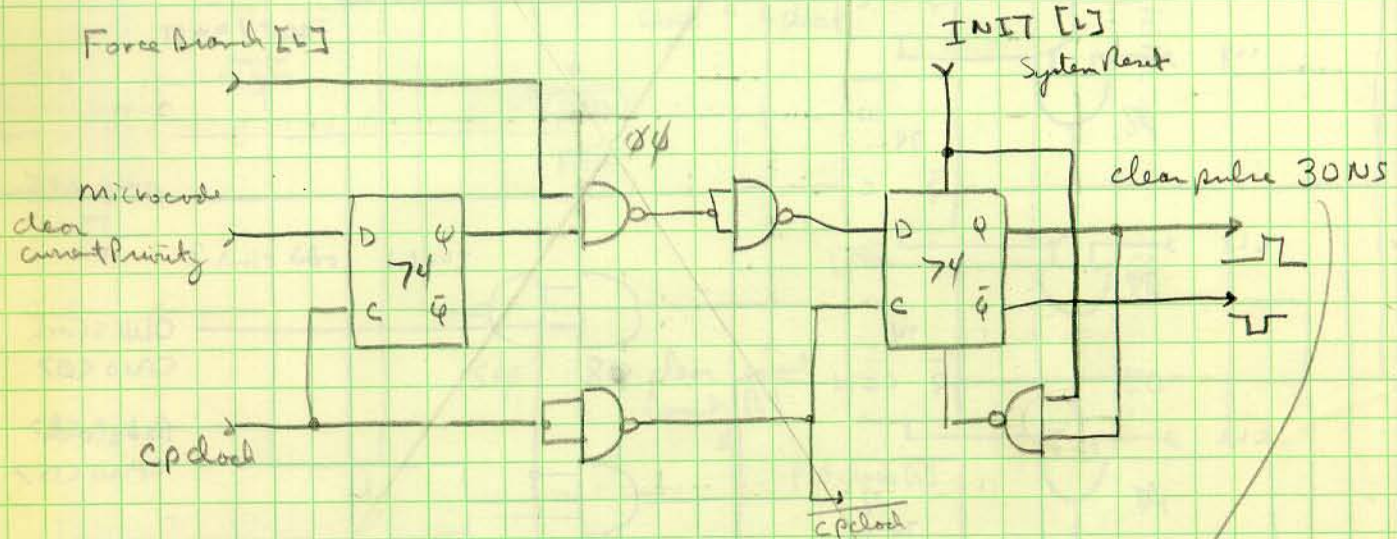
ARO

Clear Current Priority Logic

Modified to clear at mid cycle

1) don't clear current priority during a BR I CODE
   if attempted clear will cause BR I code to
   branch to an undefined location

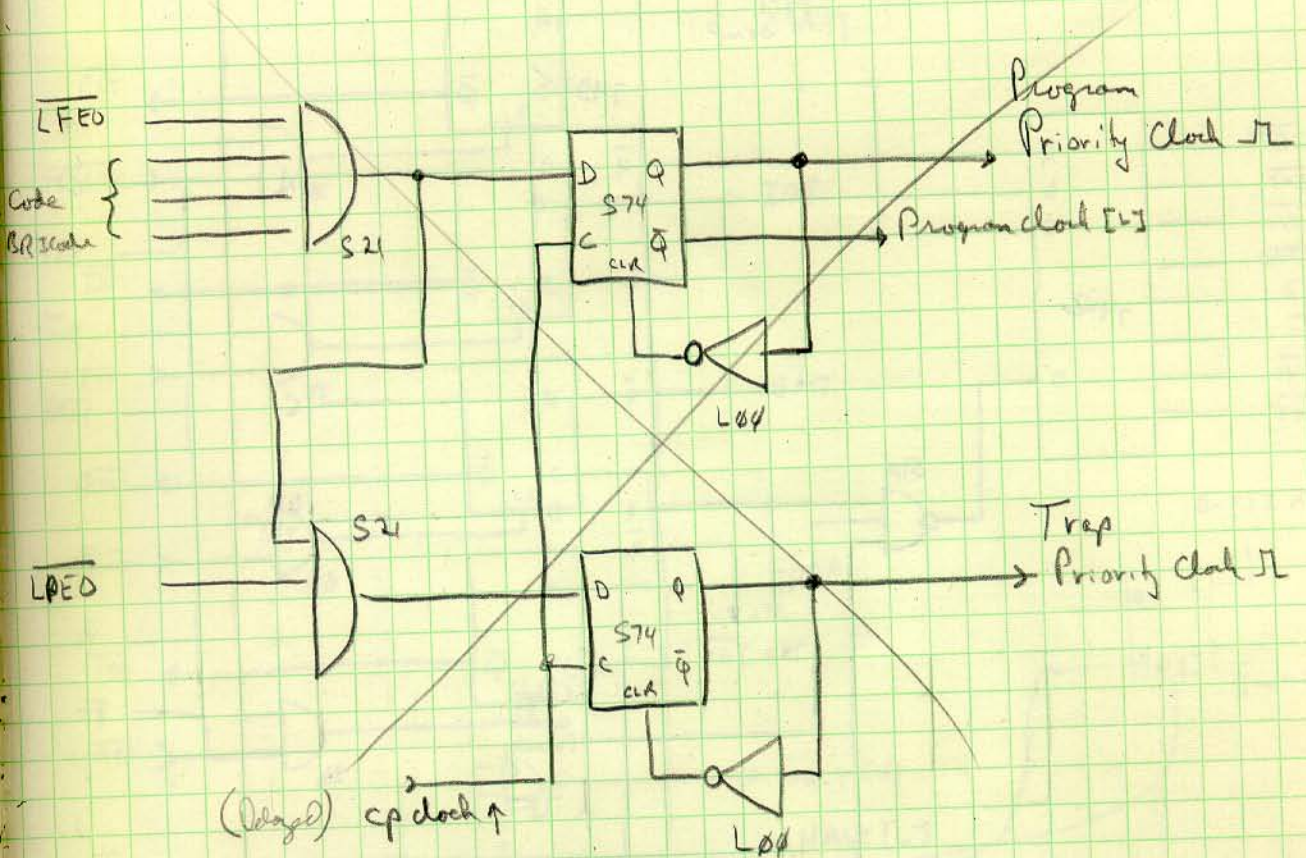
2) a clear current priority will be aborted if
   a force branch is in progress



clear priority deleted 6 June 1976
& reloaded 7 June 1976 AHP

With LPED
pulse ~ 55 NS wide

Priority Clocks (caused by BR Icode) LW



Force Priority is checked every microcycle

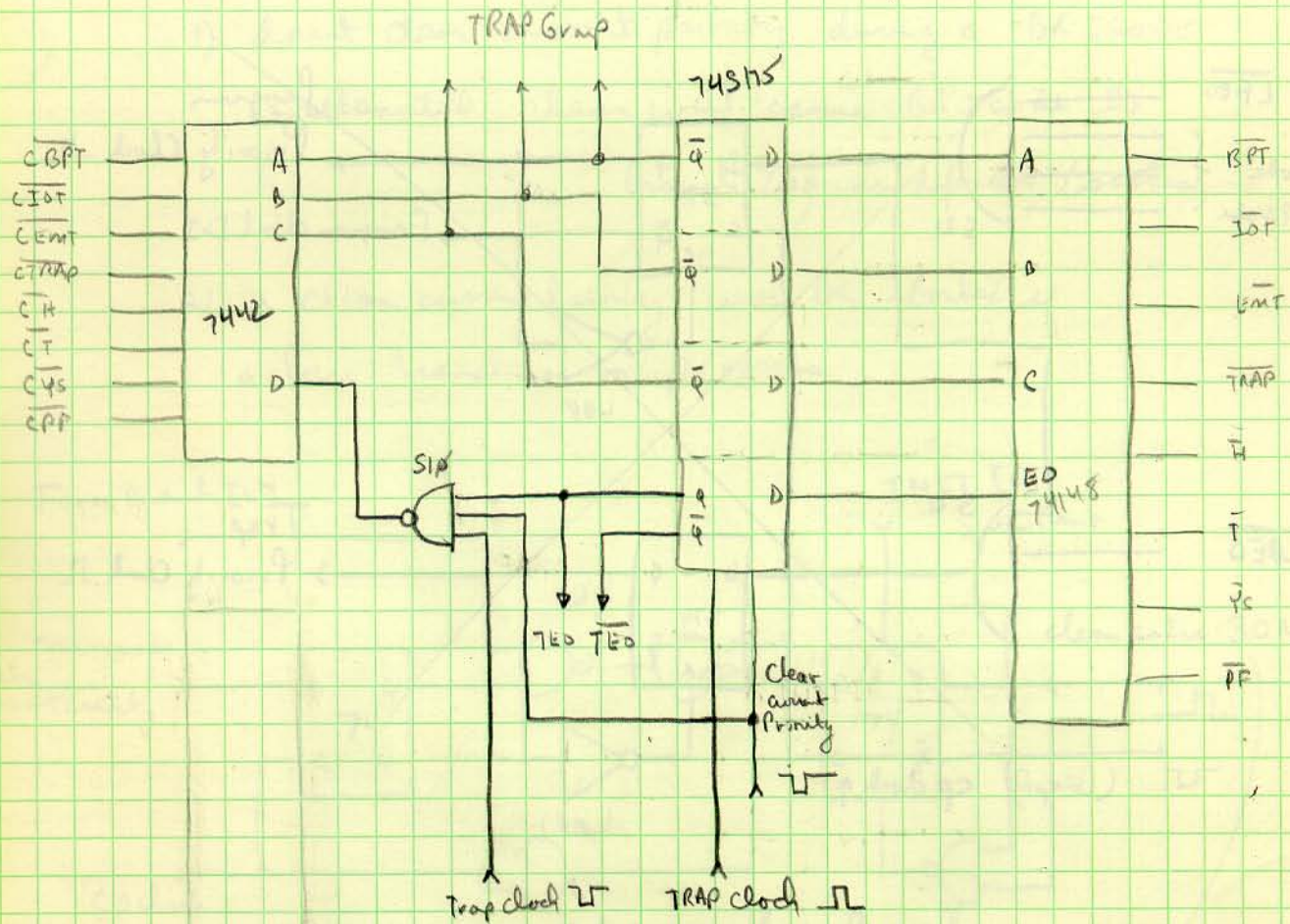
Interrupt Priority is are checked every DATI IR

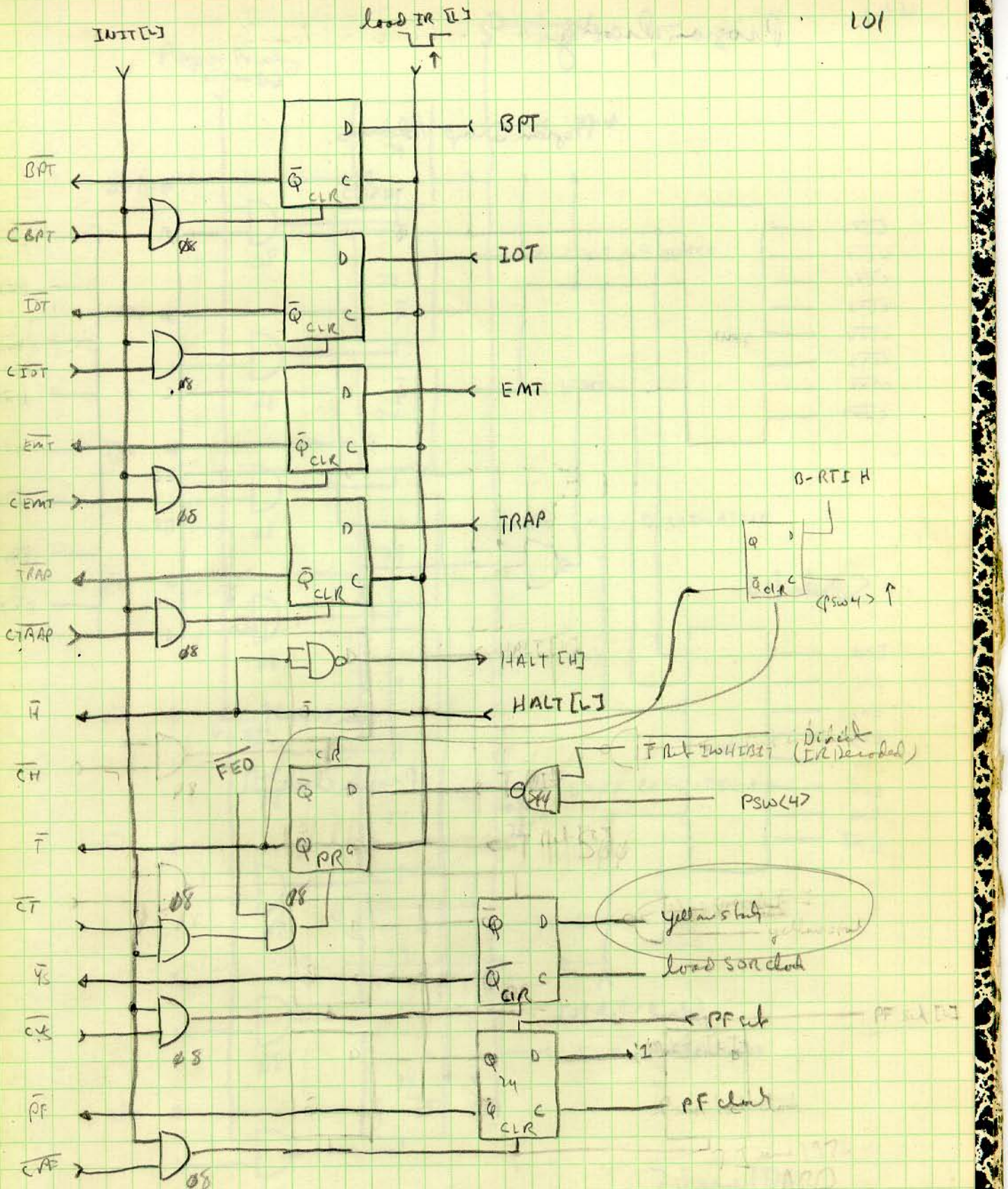
If a Force Branch is pending the Program / Trap clocks are inhibited

Redesigned 14 Jan, 76
ARO

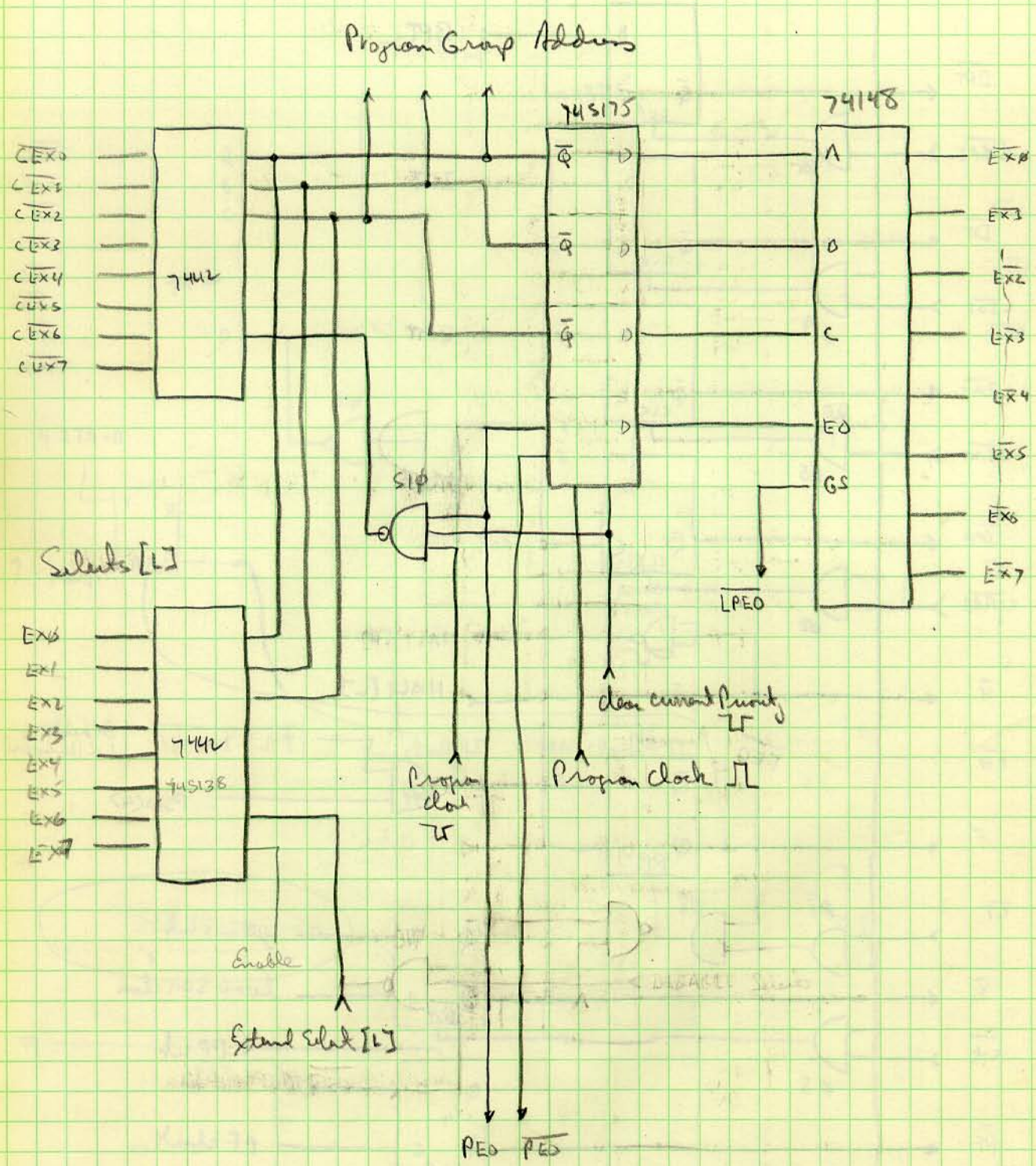
6 June 76
ARO

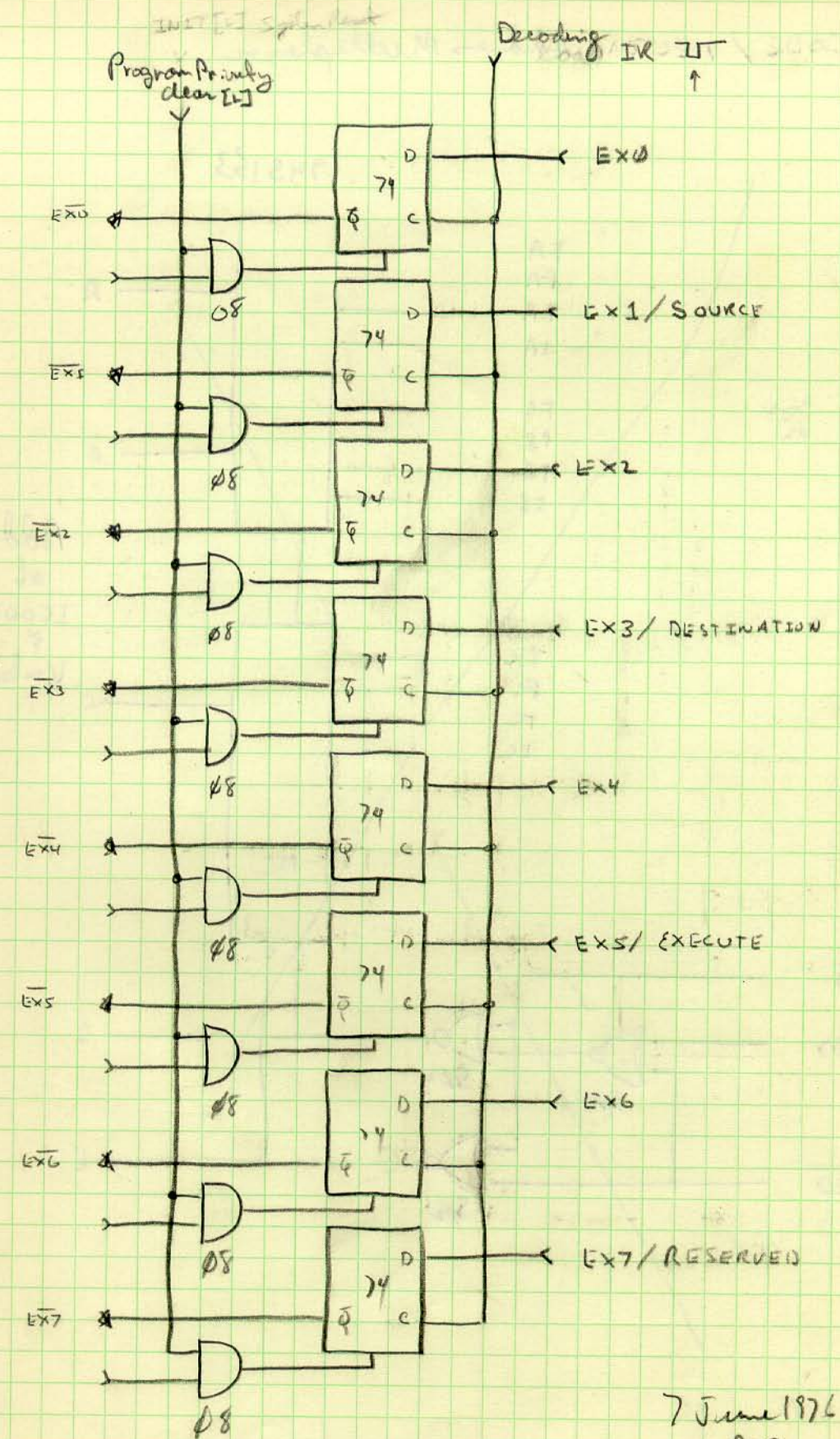
TRAP Group Priority Encoding + Addressing




INIT[L]
load IR [L]
BPT
IOT
EMT
TRAP
HALT [H]
HALT [L]
B-RTI H
(PSW 4) ↑
FEO
PSW(4)
Direct (IR decoded)
yellow stb
load SOR clock
PF set
'1'
PF clock

Program Priority





7 June 1976
ABD

BR ICODE / VECTOR Address Multiplexer