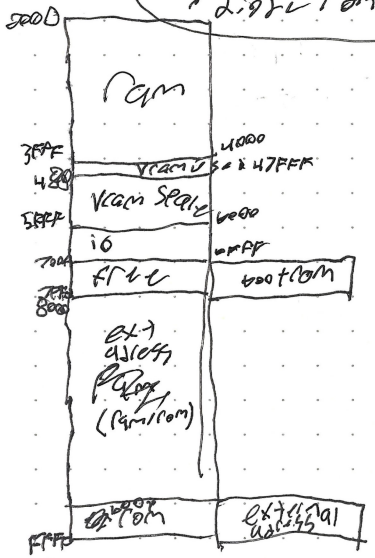
basic TTL / Discrete computer
8bit

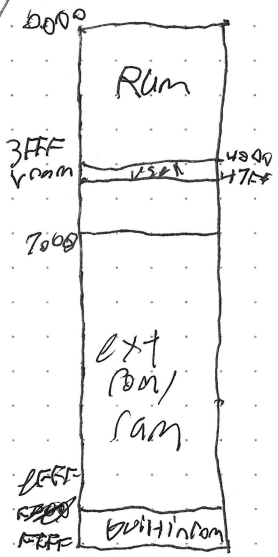
- program counter
- registers
  - Accumulator
  - B
  - C
- ALU
  - 8bit
  - Discrete gates or TTL
- MAR (memory address register)
- program memory
- instruction register
- output register
- clock
  - 555 timer

- control matrix
- digit rom?



formatter
Out

Early Spec brainstorming
and early memory map



Basic Computer - memory map / components specs

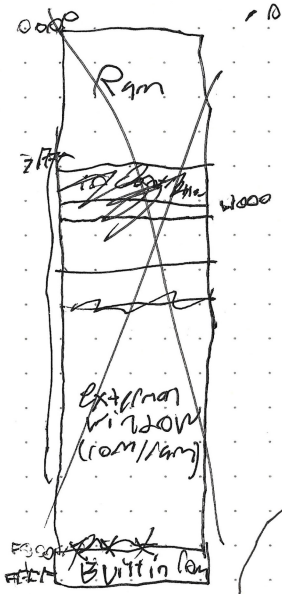
CPU
6502 or Z80

Video
MC6847 or TMS9918

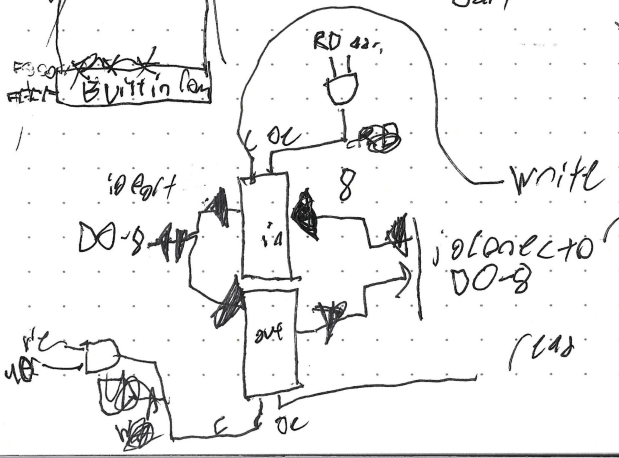
Ram
16K or 32K

IO
- KB controller or some sort
- game pad
- cassette
- gpio
- built in dot matrix screen
- Uart?
- on comp buttons

Sound
AY-3-8910 SN 76489



- 6502
- io shift reg
  - KB
  - gpio
- MC6845 CRTC
- "CPU"
- io registers?
- uart
- ascii KB
- 82C54 timer
- 16C550 uart
- ba4802YRC



MMIO

microcomputer – I/O select logic

6000–6FFF

A15 … A0
0110 xxxx xxxx xxxx

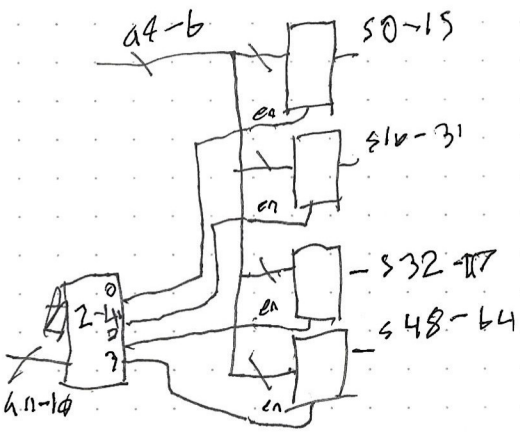
io in ~~256 banks~~
atleast 64 banks

0110 aaaa ~~bbbb~~ xxxx

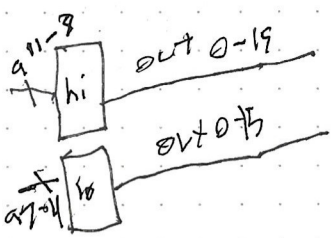
16 reg 1: 4-16 line converter
~~128~~ 64 5: 4-16 line conv

MMI/O mapping and select brainstorming

chained line conv



01



AND high nibble low nibble

microcomputer – register list

registers 00–FF (each register has 16 subregisters)

Register Mapping (Final?)

00 – MC6845 register
- subregister
  - ~~00~~: ~~register select~~ + Address register
  - *1: selected register

01 – KB register (irq generated by 21)

02 – Timer enable register 0b1 1 1 1 1 (7 6 5 4 3) PXE PIE TIE (2 1 0)
- PXE: pulse x enable 0=Disable 1=enable
- TIE: timer IRQ enable 0=Disabled 1=enabled

03 – 82C54
- 0: counter 0
- 1: counter 1
- 2: counter 2
- 3 control word register

04 – HD44780 LCD Dm

Yes
- 0: command register
- 1: data register
- 2: enable
- 3: contrast

clock toggle

0: control
1: command ~~xxxx~~ BCKLT RW E
2: contrast

1: xxxx BCKLT RS RW E

05 – button

06 – joy

07 – 16C550
- 0: RX buffer (R) / tx holding (w)
- 1: interrupt enable
- 2: interrupt ident (r) FIFO control (w)
- 3: Line control
- 4: modem control
- 5: line stat
- 6: modem stat
- 7: scratch

When bit set → 0: Div Latch (LSB) 1: Divisor Latch (MSB)

7F – mapping (timer/IRQ) Dirtch
- 0: CHRam
- 1: video window = BROM
- 2: video window = CHRam
- 3 enables

CHRS: CHRam | VBS: VRAM | BS0 | BS1 | BS2

BS0 (FX Page) 0=Bootrom 1=ram

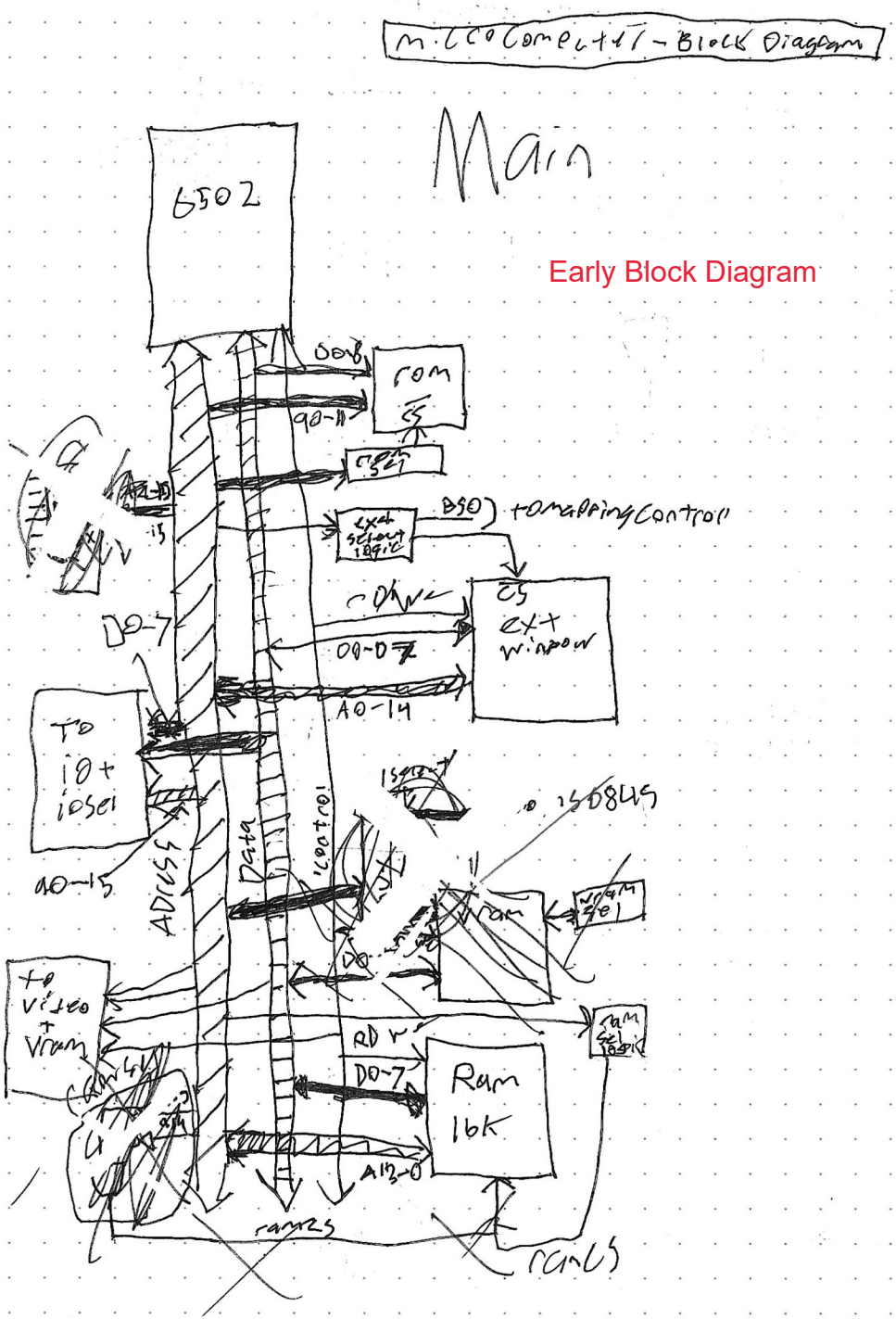
BS1 (7X page) 0=bootrom 1=nothing

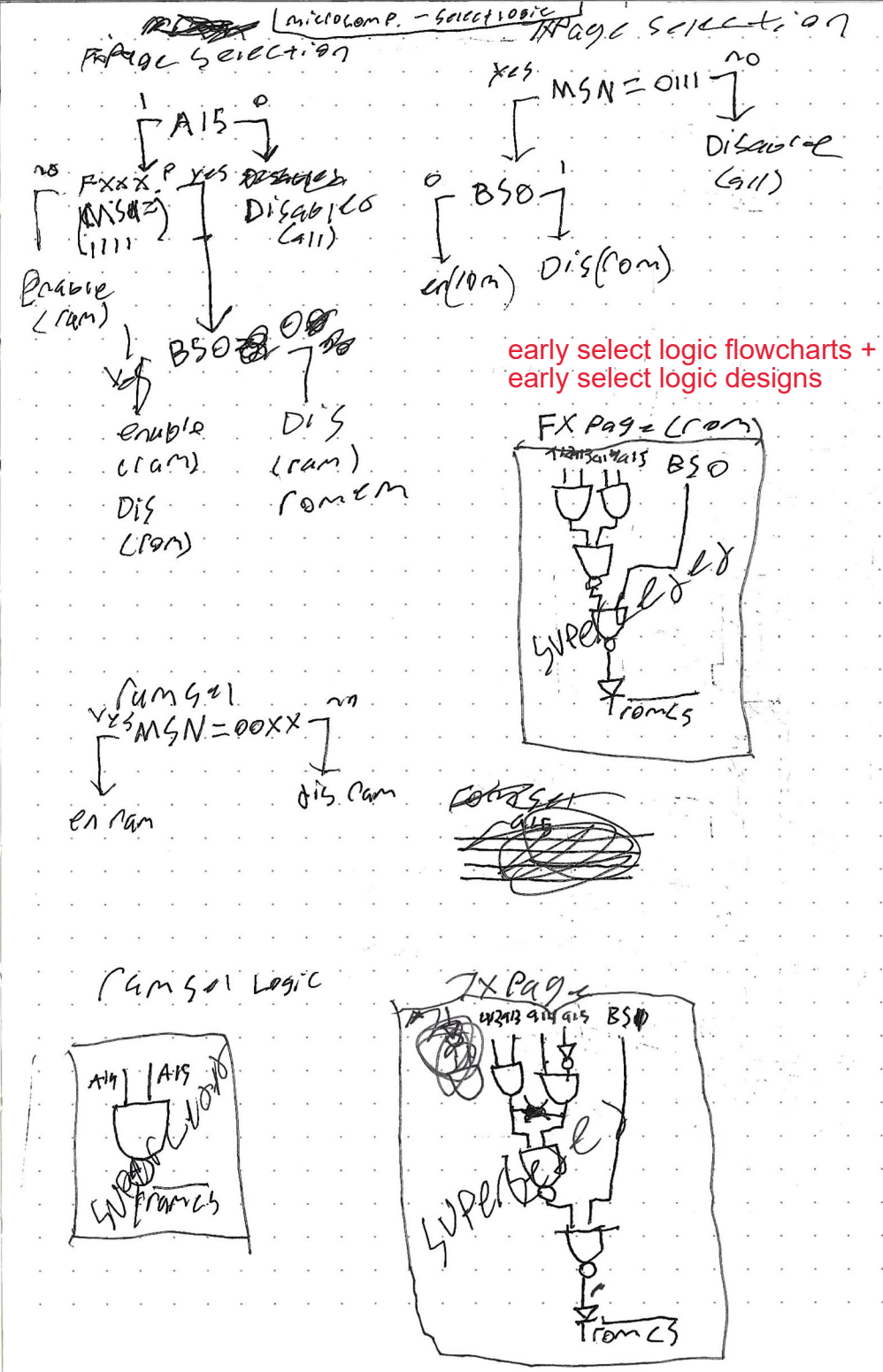
BS3 rambank VRAM bit

BS2 rambank lower bit

00 ix updates → VBS: VRAM bank select
LPV application 0=bank0 1=bank1

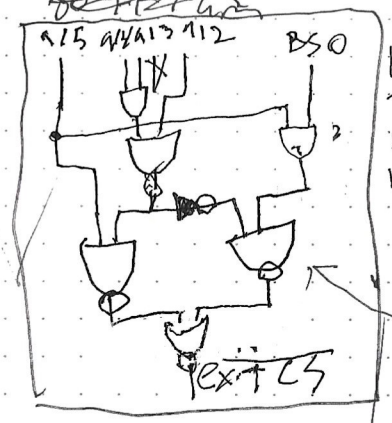
80–FF reserved


M.LCO Computer - Block Diagram
Main
Early Block Diagram
6502
rom
CS
ext select logic
BS0) +mapping control
CS
ext window
to io+ iosel
ADRESS
Data
control
ram
to video + vram
Ram 16k
A13-0
ramCS


microcomp. - selectlogic
Page selection
A15
Disable (all)
BS0
enable (ram)
Dis (ram)
romCS
MSN = 0111
Disable (all)
BS0
en(rom)
Dis(rom)
early select logic flowcharts +
early select logic designs
FX Page (rom)
BS0
romCS
MSN = 00XX
en ram
dis ram
ramsel Logic
A14
A15
ramCS
FX Page
BS0
romCS

microcomp. - select logic cont.

ext select



1 and
3 nand
1 not
1 nor

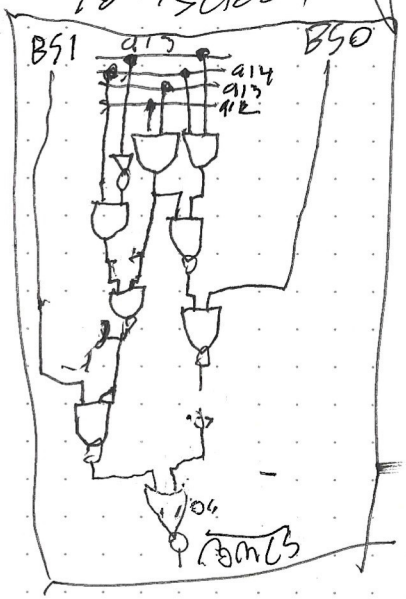
io select

~~2 not~~
~~2 and~~
~~1 nand~~

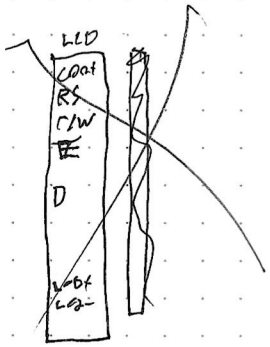
works when a15=0

More select logic
(some final, some not)

rom select



3 and
1 not
4 nand
1 nor

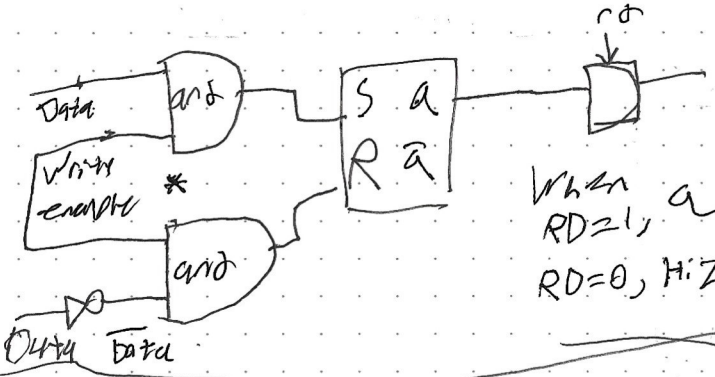


1 nand
2 and

~~8 and~~
~~4 not~~
~~9 nand~~

9 nand
7 and
3 nor
2 not

11 nand
~~8~~ and
3 nor



When RD=1, a
RD=0, Hi Z

microcomp. - registers + select logic cont.

Registers cont

08 RTC

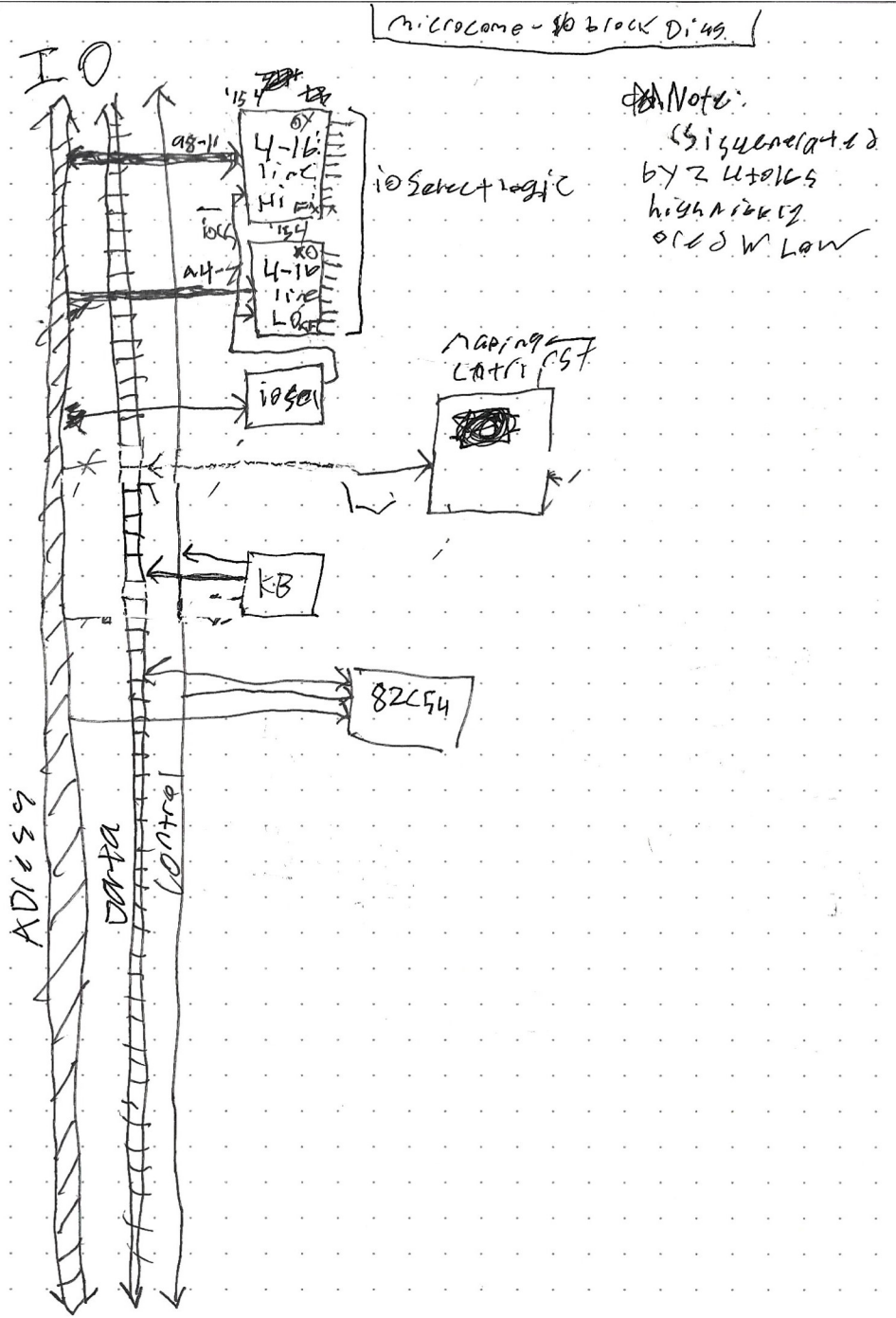
0 - sec
1 - sec alarm
2 - min
3 - min alarm
4 - Hours
5 - Hours alarm
6 - Day
7 - Day alarm
8 - Day of week
9 - month
A - year
B - ram
C - Enables
D - Flags
E - control
F - Century

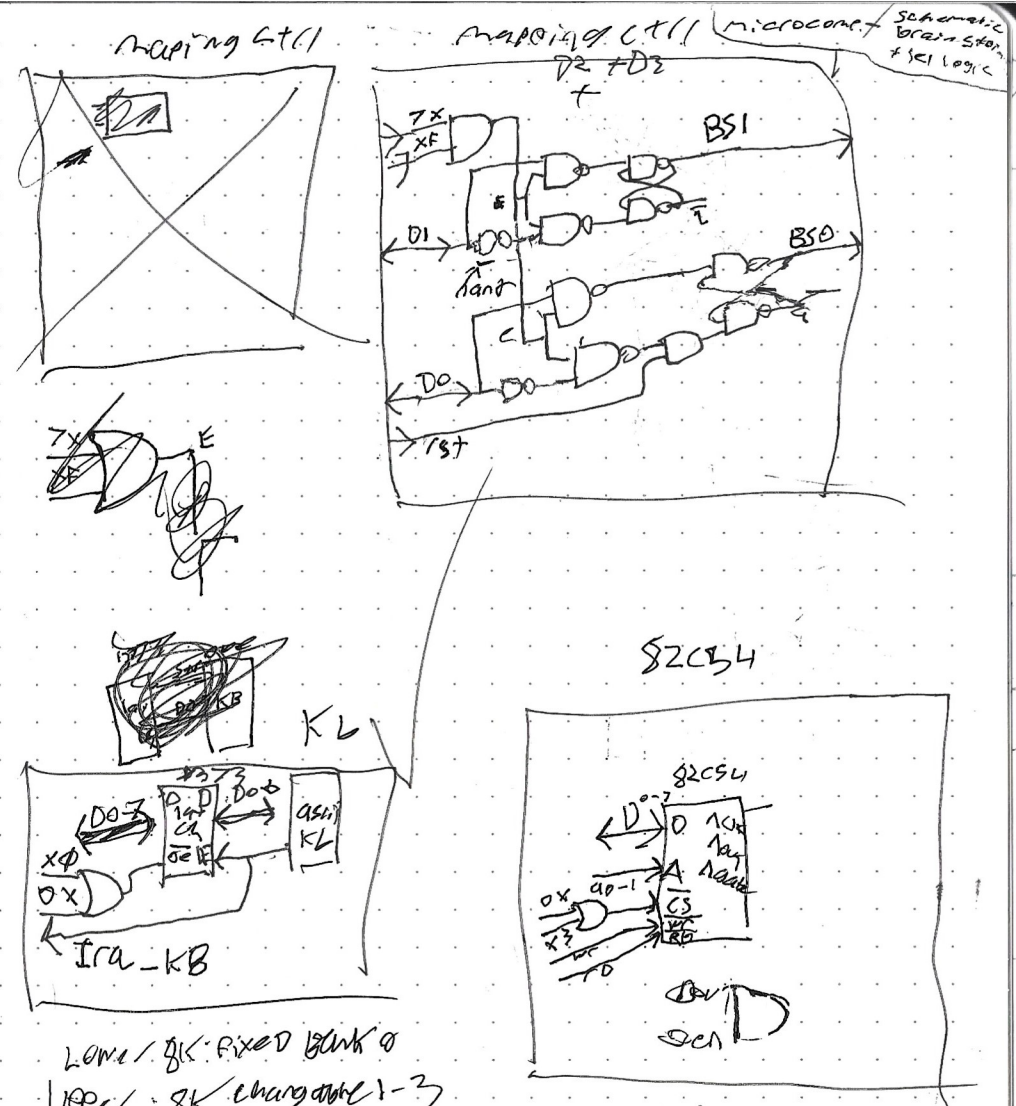
RTC Registers (scrapped) +
final(?) I/O Select signal generation

a	b	y
0	0	1
1	0	0
0	1	0
1	1	0

io select

1 and
1 nand
1 nor





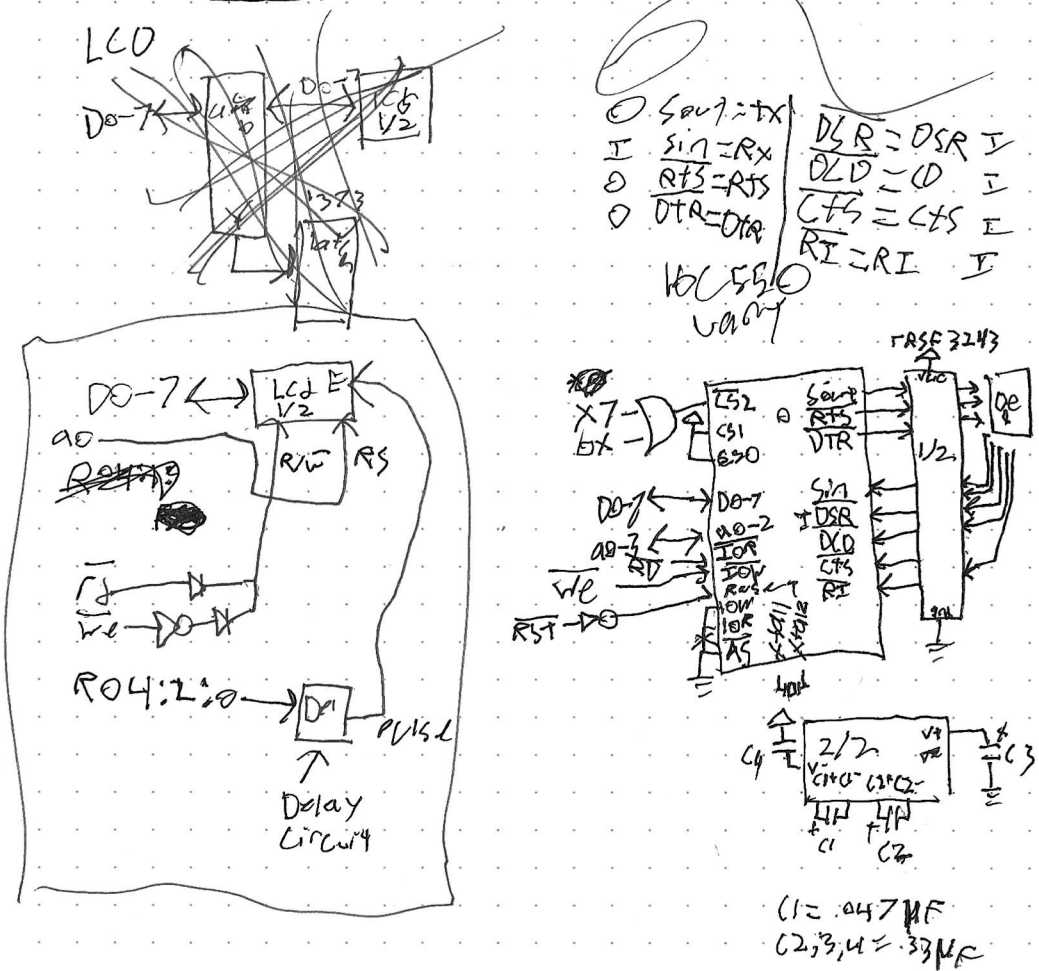
Early idea for mapping control logic (superceded)

Keyboard brainstorming

Ram select logic

82c54 Brainstorming

microcomp - Schematic Brainstorm



LCD Brainstorming (scrapped)
RS232 Schematic Brainstorming

Idea for built in toggleable null modem adapter (Scrapped)

internal port

CD a pin1 a
b pin6 b

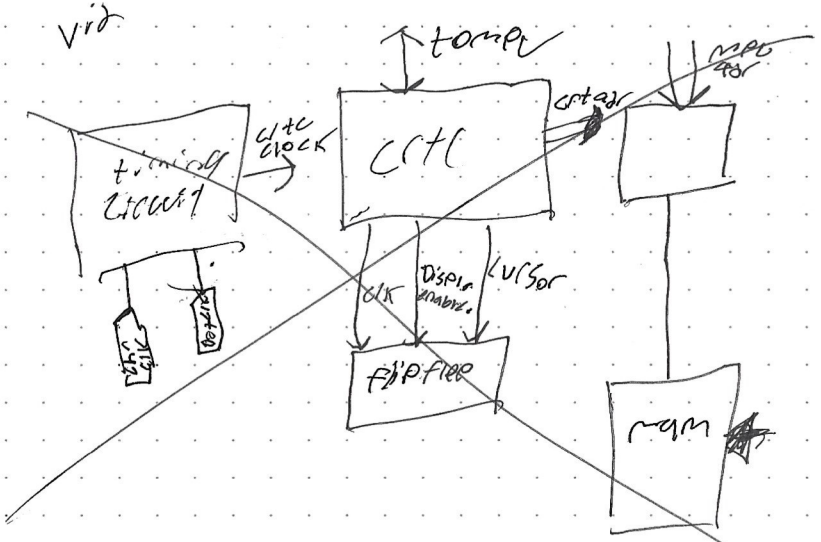
CD (internal) → CD pin1 (ext)
DSR (pin6)

pin1 (CD ext) → DTR pin4

When a, CD (internal) connected to CD (ext) pin1

When B, CD (int) connected to DSR pin6 AND CD (ext) pin1 to DTR

Video



Built in Rom microcomp - rom brainstorm = Optional

- WOZMON (save prg
- LCDDriver ✓
- IRQ routine ✓
- Button Driver ✓
- KB Driver
- 6545 Driver
- 16C550 Driver ✓
- RS232 ram upload prg

Final Rom Brainstorming

* First instruction is to disable interrupts

micro – todo

- ✓ video
  - ✓ vram logic
  - ✓ Timing
  - ✓ video logic

Gpio 8bit

- CPU stuff
  - Interrupt handling all together
  - timing

Joy: 8bit

- I/O
  - LCD Screen
  - joy
  - nav buttons] uses 65C22
  - gpio — uses 65C21
  - finish timer IC – Close: need to implement speaker and
  - RTC

PICsbit SERVO

Todo

- via for joy/buttons/ gpio
- via for KB? (ascii might overwhelm or a server)

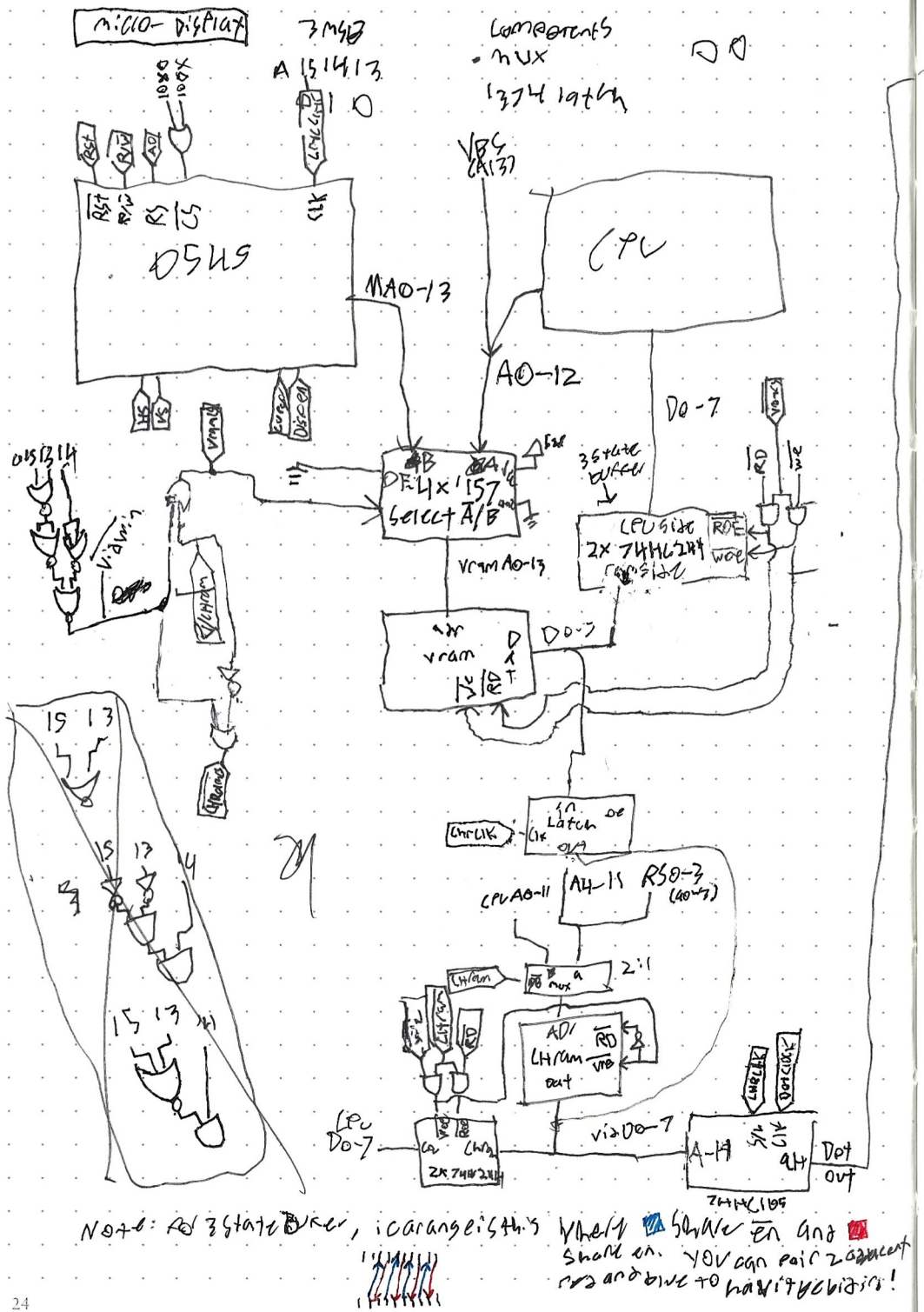
---

micro-ina check

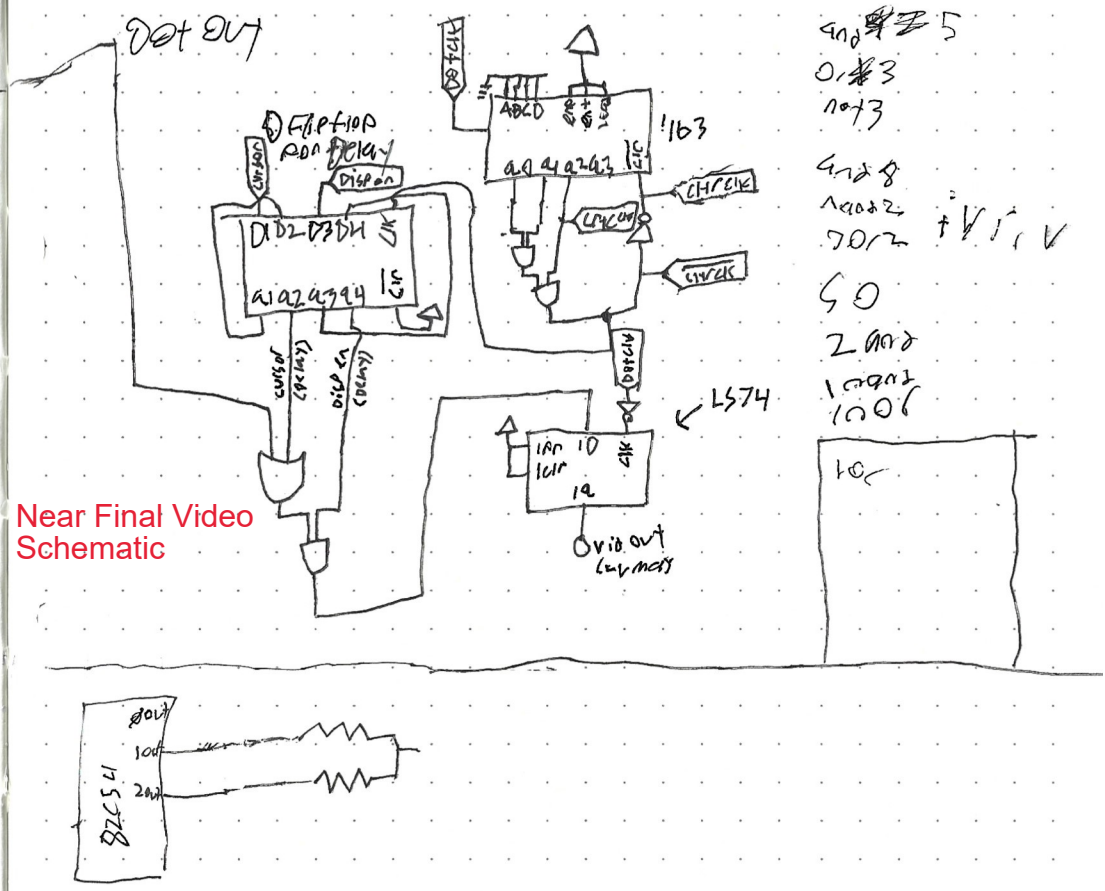
- 82C54 (can must be analog format)
  1. write 1 to correct CNT bit(s) 03:3:3-1
  2. write 0 to status 03:3:4
- KB (check registers on 65C21)
- 16C550
  - IRQ must be inverted and anded

IRQ Checking routine Brainstorm

ALWAYS BCKP



Note: A0 3state buffer, i can change it this where ■ shorter en and ■ short en. You can pair 2 adjacent rows and drive to half the display!



Final Mapping + Registers

PCB Layout + notes
1 CPU
2 CPU
3 CPU
4 Sel Logic
5 Sel logic
6 Sel Logic + Reset inverting
7 Sel Logic
8 Sel Logic
9 Sel logic
10 io sel logic
11 IO Sel logic
12 register (rcvr io sel)
13
14 timer
15 KB
16 KB + Serial + gate, or + Timer (io sel logic)
17 Serial
18 Serial
19 via
20 GPIO
21 Sel logic
22 register
23 register
24 via
25 via CHram
26 via
27 via } Vram
28 via
29 via (timing) + via (combi) + Timer
30 via (sel logic) + io sel
31 via (sel logic)
32 via (sel) + via (timing) + via (combi) + Serial + timer
33 via (vram)
35 via (vram)
36 via (vrams)
37 via (vram) + via (CHram)
38 via (vram) + via combi
39 via
40 via } CHram
41 via
42 via } CHram
43 via
44 via (timing)
45 via (delay)
46
47 GPIO
48 GPIO
49

26

Fix Chram Buffer rounds

tmux

note 1

7	6	5	4	3	2	1	0
JOYSEL					P2E	P1E	TIE

P2E Pulse X enable 0 disable 1 enable
P1E
TIE Timer interrupt enable 0 disable 1 enable

note 2

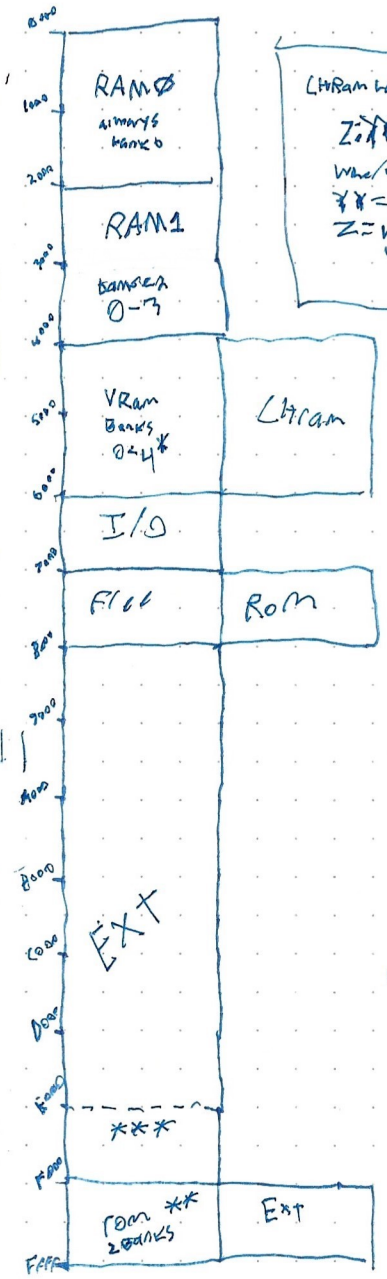
7	6	5	4	3	2	1	0
VICHRAM	CHBS	VBS1	VBS0	BS3	BS2	BS1	BS0

BS0: ram Bank switch FX Page 0: Boot rom 1: RAM
BS1: rom Bank switch 7X Page 0: Boot rom 1: nothing
BS2-3: RAM 1 ram Bank 0-3
VBS0: vram Bank select 0: lower 1: upper * affects CPU only
VBS1: vram Bank select 0=Bank0 1=Bank1 * affects CPU and crtc
CHBS: CHram Bank select 0=Bank0 1=Bank1 * affects crtc
VICHRAM: selects which areas in vid sel win 0: vram 1: CHram

Note 3: Function depends on whether or not LCR bit 7 (DLAB) is set

DLAB = 0: TX/RB Buffer And Interupt Enable Register (IER)
DLAB = 1: Divisor Latch

io map

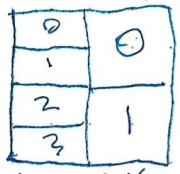


Ram 32K
VRam 32K
CHram 8K

CHram layout
Z:YY:X
where X = row num
YY = CH/ num
Z = which CHram bank

Area's located at
0:0Z YYYY:YYYYXXXX

* Vram but appears to the crtc as 2



CHram appears as bank +0 CPU but 2 +0 crtc

** Ram is split into 2 banks can be controlled in 3 ways

a. manual (controls by system)
b. register
c. ~~****~~ (xpec) non ~~[illegible]~~ so that both banks are visible

Registers

CRTC 00:0 register select
00:1 selects register

KB 01 Keyboard

Reg 02 Timer Enable Register (note 1)

Reg 03:0 Counter 0
03:1 Counter 1
03:2 Counter 2
03:3 Control word register
04 - Reserved -
05 - Reserved -
06 - Reserved -

UCSR 07: 0 TX/RX ~~[illegible]~~ Buffer / Divisor Latch LSB
1 interrupt enable / Divisor Latch MSB (Note 3)
2 interrupt identification
3 Line control
4 modem control
5 line stat
6 modem stat
7 Scratch

Reg 7F: Bank select (note 2)

27

Video Calculations

mc6845 Calculations for
8x8 40x24 char video
+ video mode ideas (postponed)

Vid modes
00: 8x8 40x24 10: 8x16 40x24
01: 8x8 80x24 11: 8x16 80x24

B1 Hor. freq. B6 # disp char row

B2 Vert freq B7 dots in char dot mtx.

B3 min. hor. retrace time B8 # scanline in char dot mtx row R_n = register

B4 min. vert. retrace time B9 # dots between adj char

B5 # disp. char per row B10 # scanlines

Xtal freq = 6.174

Vert ≈ 60Hz!

too little time, mode

$F' = \frac{B_5 \cdot (B_7 + B_4)}{\left(\frac{1}{B_1}\right) - B_3}$

(dot freq. (15.4 approx.))

t_c char time $= \frac{1}{(R_0 + 1) \cdot B_1}$

$R_7 = 32 - \frac{16-6}{8} \geq R_7 \geq R_6$

$30.75 \geq X \geq 6$

So $R_7 = 7$

F dot freq very $\frac{B_7 + B_9}{T_c}$

t_{SL} time scanline $(R_0 + 1) \cdot T_c$

n total # scanlines

$t_{hr} \frac{(48+1-40) \cdot 8}{6.174 \times 10^6} = 1.6618 \times 10^{-5}$

$t_{hr} > B3$? Yes!

mode 00

B1: 15750

B2: 60

B3: 1×10^{-5}

B4: 9.5×10^{-4}

B5: 40

B6: 8

B7: 8

B8: 8

B9: 0

B10: 0

$F' = \frac{40 \cdot (8+0)}{\left(\frac{1}{15750}\right) - 1 \times 10^{-5}} = 5.982$ MHz

$t_c \frac{1}{40 \cdot 15750} = 1.2957 \times 10^{-6}$

$F \frac{8}{1.2957 \times 10^{-6}} = 6.174 \times 10^6$ (MHz)

t_{SL} $40 \cdot 1.2957 \times 10^{-6} = 6.3492 \times 10^{-5}$

$n \frac{1}{60 \cdot 6.3492 \times 10^{-5}} = 262$

$\frac{262}{8} = N + \frac{R}{8}$ $32 + \frac{6}{8}$ $n = 32$ $R = 6$

R_1 $8 - 1 = 7$

t_{LA} $8 \cdot 6.3492 \times 10^{-5} = 5.0794 \times 10^{-4}$ 50.8 μs

R_0 $\frac{5.9822 \times 10^6}{15750 \cdot 8} = 48$

$R_1 = 40$

R_2 $40 + \frac{3}{2} = 42$

R_3 $\frac{48-40}{3} = 3$

$R_4 = 31$

$R_5 = 6$

$R_6 = 6$

$R_7 = 7$ (7 − 30) OK

$R_8 = 7$

$\left(\frac{15750}{60} - 8(8)\right) \cdot 6.3492 \times 10^{-5}$

$t_{VR} = 1.260 \times 10^{-2}$

$t_{VR} > B_4$? Yes!

Mode 01

B1: 15750

B2 60

B3 1×10^{-5}

B4 4.5×10^{-4}

B5 40

B6 24

B7 8

B8 8

B9 0

B10 0

~~B15 Diffracted~~ reg6

bb Dik ~~R0~~ Dr

R0: 48

R1: 40

R2: 42

R3: 3

R4: 31

R5: 6

R6: 24

R7: 28

Clock xtal is OK

30 > 28 > 24

rescale

t_{ra}

R6

R7

$F' = 5.482$ MHz

$t_c = 1.2457\ldots\times10^{-6}$ s

$F = 6.174$ MHz