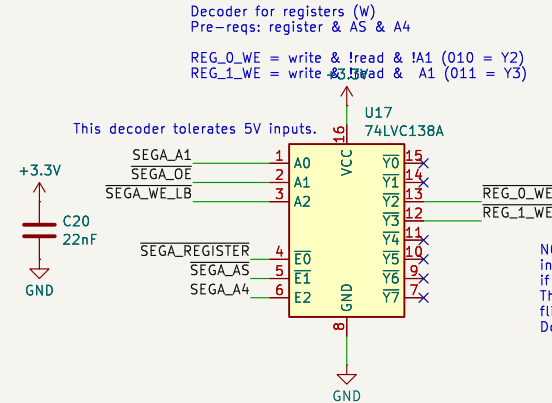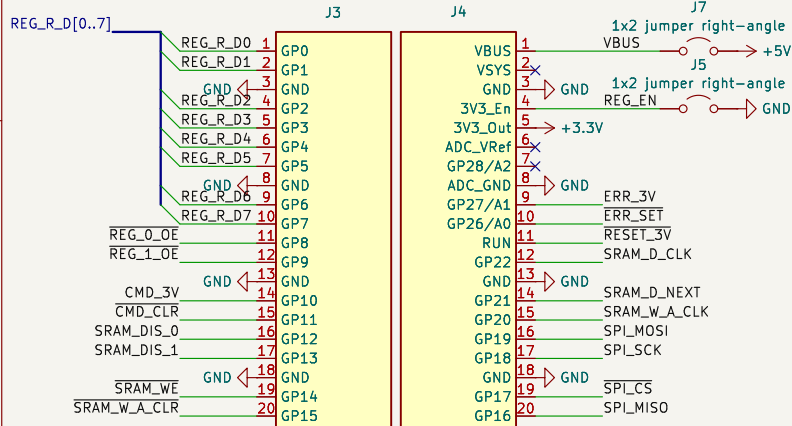The 3.3V regulator built into the microcontroller is documented as
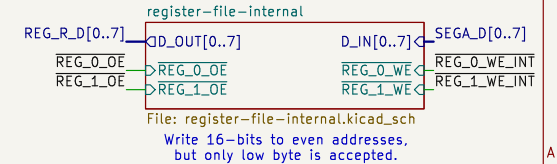only supplying a max of 300mA to external circuitry.

Rather than use that, we use our own regulator in the cartridge board.
It can power the microcontroller and all 3V logic chips. Then we power
the microcontroller through the 3V3_OUT and GND pins.

One jumper connects 3V3_EN connects to GND to disable the board's
regulator. Another jumper connects VBUS to the +5V supply to power
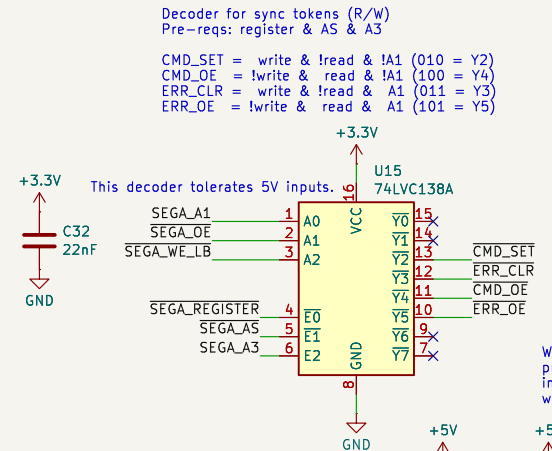the 5V chips.

While programming the microcontroller in-place via USB, the jumper should
be on the VBUS pins. While operating in the Sega, the jumper should be
on the 3V3_EN pins.

Decoder for registers (W)
Pre-reqs: register & AS & A4

REG_0_WE = write & !read & !A1 (010 = Y2)
REG_1_WE = write & !read & A1 (011 = Y3)

This decoder tolerates 5V inputs.

U17
74LVC138A

SEGA_A1   1   A0   VCC 16
SEGA_OE   2   A1   Y0 15
SEGA_WE_LB   3   A2   Y1 14
                              Y2 13   REG_0_WE
                              Y3 12   REG_1_WE
SEGA_REGISTER 4 E0   Y4 11
SEGA_AS   5   E1   Y5 10
SEGA_A4   6   E2   Y6   9
                      GND   Y7   7
                          8

+3.3V
C20
22nF
GND
GND

register-file-internal

REG_R_D[0..7]   D_OUT[0..7]   D_IN[0..7]   SEGA_D[0..7]
REG_0_OE   REG_0_OE   REG_0_WE   REG_0_WE_INT
REG_1_OE   REG_1_OE   REG_1_WE   REG_1_WE_INT

File: register-file-internal.kicad_sch

Write 16-bits to even addresses,
but only low byte is accepted.

Register addresses
CMD: 0xA13010
ARG: 0xA13012

Command token address
0xA13008

Error token address
0xA1300A

NOTE: Because of limitations in the number of inputs
in our address decoders, a conflict is possible
if the addresses 0xA13018 - 0xA1301F are written.
This range could trigger both a sync token to be
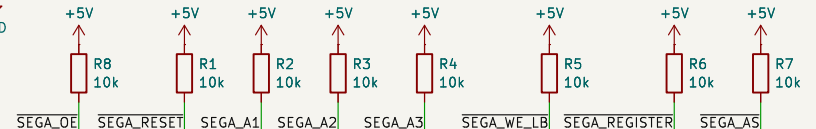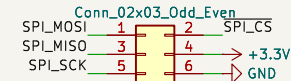flipped and a register to be written at the same time.
Don't do it.

sync-token-internal

CMD_SET   CMD_SET   CMD_3V_ALWAYS_ON   CMD_3V
CMD_CLR   CMD_CLR   SEGA_D0   SEGA_D0
CMD_OE   CMD_OE   ERR_3V_ALWAYS_ON   ERR_3V

ERR_SET   ERR_SET
ERR_CLR   ERR_CLR
ERR_OE   ERR_OE

File: sync-token-internal.kicad_sch

Decoder for sync tokens (R/W)
Pre-reqs: register & AS & A3

CMD_SET = write & !read & !A1 (010 = Y2)
CMD_OE = !write & read & !A1 (100 = Y4)
ERR_CLR = write & !read & A1 (011 = Y3)
ERR_OE = !write & read & A1 (101 = Y5)

+3.3V
C32
22nF
GND

This decoder tolerates 5V inputs.

U15
74LVC138A

SEGA_A1   1   A0   VCC 16
SEGA_OE   2   A1   Y0 15
SEGA_WE_LB   3   A2   Y1 14
                              Y2 13   CMD_SET
                              Y3 12   ERR_CLR
SEGA_REGISTER 4   E0   Y4 11   CMD_OE
SEGA_AS   5   E1   Y5 10   ERR_OE
SEGA_A3   6   E2   Y6   9
                      GND   Y7   7
                          8
GND

These bridges can be used to snoop on these signals,
or cut with a knife to hijack them.

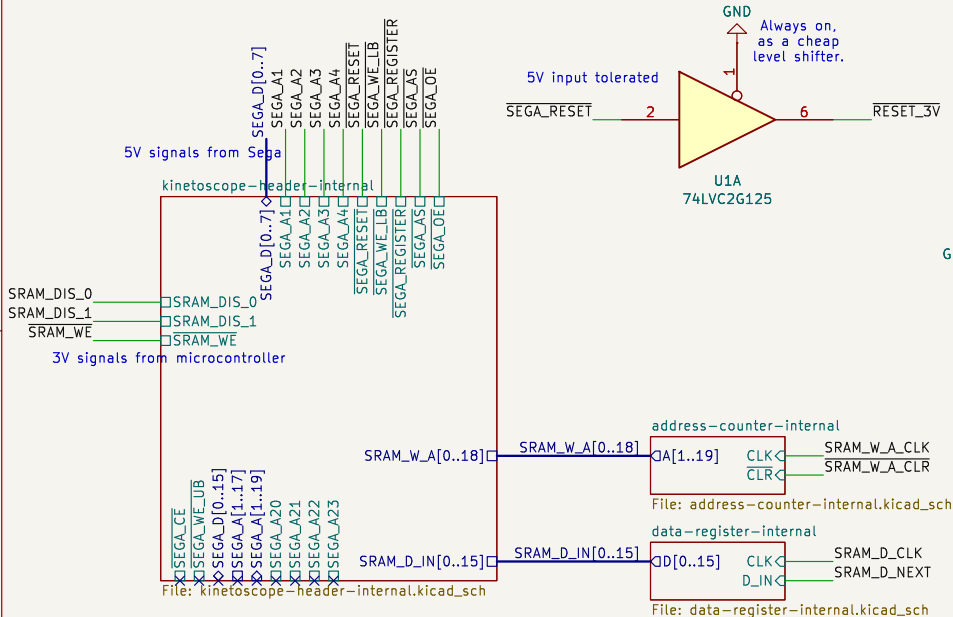REG_0_WE   J_WE_0   REG_0_WE_INT
REG_1_WE   J_WE_1   REG_1_WE_INT

While powering the microcontroller via USB without the Sega (during
programming and testing), pull-up resistors on these lines keep
inputs from floating. They can be set low for testing with jumper
wires in the cartridge pin header.

REG_R_D[0..7]

J3   J4   J7
                                                 1x2 jumper right-angle
REG_R_D0 1   GP0   VBUS 1   VBUS   J5
REG_R_D1 2   GP1   VSYS 2   1x2 jumper right-angle
GND   3   GND   GND   3   GND
REG_R_D2 4   GP2   3V3_En 4   GND   REG_EN
REG_R_D3 5   GP3   3V3_Out 5   +3.3V
REG_R_D4 6   GP4   ADC_VRef 6
REG_R_D5 7   GP5   GP28/A2 7
GND   8   GND   ADC_GND 8   GND
REG_R_D6 9   GP6   GP27/A1 9   ERR_3V
REG_R_D7 10 GP7   GP26/A0 10   ERR_SET
REG_0_OE 11 GP8   RUN   11   RESET_3V
REG_1_OE 12 GP9   GP22   12   SRAM_D_CLK
GND   13 GP10   GND   13 GND SRAM_D_NEXT
CMD_3V   14 GP11   GP21   14   SRAM_W_A_CLK
CMD_CLR   15 GP12   GP20   15   SPI_MOSI
SRAM_DIS_0 16 GP13   GP19   16   SPI_SCK
SRAM_DIS_1 17 GP14   GP18   17   SPI_CS
GND   18 GP15   GND   18 GND SPI_MISO
SRAM_WE   19 GP14   GP17   19
SRAM_W_A_CLR 20 GP15   GP16   20

GND
Always on,
as a cheap
level shifter.

5V input tolerated
SEGA_RESET   2   6   RESET_3V

U1A
74LVC2G125

+5V +5V +5V +5V +5V +5V +5V +5V

R8   R1   R2   R3   R4   R5   R6   R7
10k 10k 10k 10k 10k 10k 10k 10k

SEGA_OE SEGA_RESET SEGA_A1 SEGA_A2 SEGA_A3 SEGA_WE_LB SEGA_REGISTER SEGA_AS

+3.3V
GND
Unused gate.
U1B
74LVC2G125

Connector for optional Adafruit Ethernet featherwing
(jumper wires or adapter needed)
J6
Conn_02x03_Odd_Even

SPI_MOSI   1   2   SPI_CS
SPI_MISO   3   4   +3.3V
SPI_SCK   5   6   GND

X2
Template for RPiPicoW
board header alignment

5V signals from Sega

kinetoscope-header-internal

SEGA_D[0..7]
SEGA_A1
SEGA_A2
SEGA_A3
SEGA_A4
SEGA_RESET
SEGA_WE_LB
SEGA_REGISTER
SEGA_AS
SEGA_OE

SRAM_DIS_0   SRAM_DIS_0
SRAM_DIS_1   SRAM_DIS_1
SRAM_WE   SRAM_WE

3V signals from microcontroller

SEGA_CE
SEGA_WE_UB
SEGA_D[0..15]
SEGA_A[1..17]
SEGA_A[1..19]
SEGA_A20
SEGA_A21
SEGA_A22
SEGA_A23

SRAM_W_A[0..18]   SRAM_W_A[0..18]

address-counter-internal
A[1..19]   CLK   SRAM_W_A_CLK
           CLR   SRAM_W_A_CLR

File: address-counter-internal.kicad_sch

data-register-internal
D[0..15]   CLK   SRAM_D_CLK
           D_IN SRAM_D_NEXT

File: data-register-internal.kicad_sch

File: kinetoscope-header-internal.kicad_sch

+3.3V
74LVC2G125
C1
22nF
GND
GND VCC U1C
GND

Sheet: /
File: microcontroller.kicad_sch

Title: Kinetoscope Microcontroller and Register Board

Size: A4   Date: Rev:
KiCad E.D.A. 8.0.6   Id: 1/6

J1

Pin	Left Label	#		Right Label	#	Out Label

SEGA_A1
SEGA_A2
SEGA_A3
SEGA_A4
SEGA_A[1..19]
SEGA_A[1..17]

◇ SEGA_D[0..7]
◇ SEGA_D[0..15]

J1
SEGA_A1 1 SEGA_A1   SEGA_D0 25 SEGA_D0
SEGA_A2 2 SEGA_A2   SEGA_D1 26 SEGA_D1
SEGA_A3 3 SEGA_A3   SEGA_D2 27 SEGA_D2
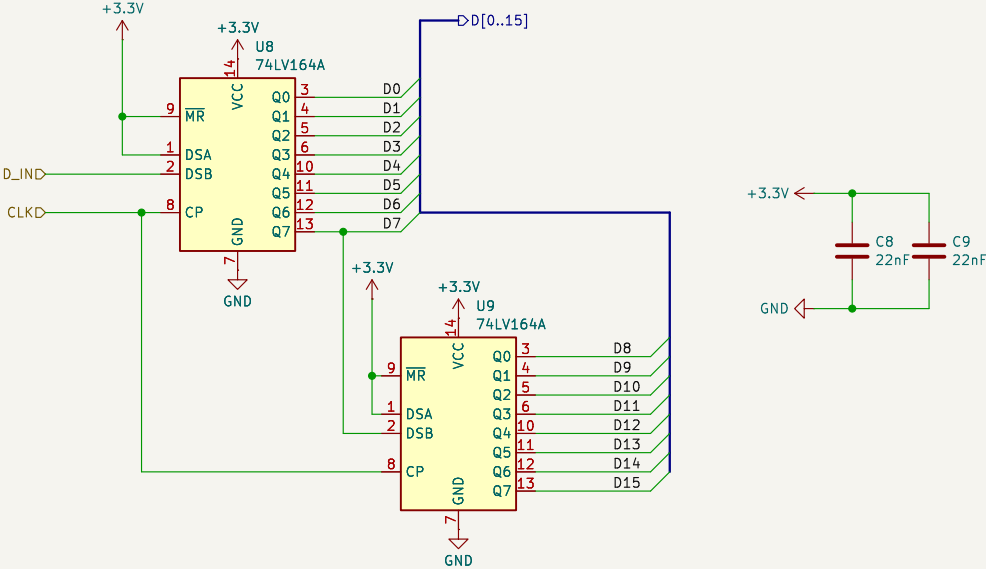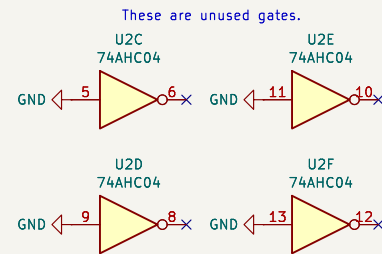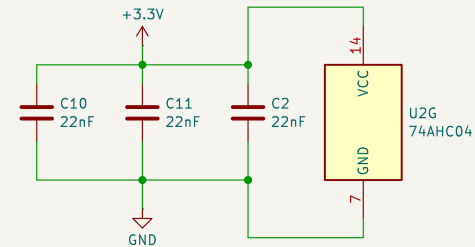SEGA_A4 4 SEGA_A4   SEGA_D3 28 SEGA_D3
SEGA_A5 5 SEGA_A5   SEGA_D4 29 SEGA_D4
SEGA_A6 6 SEGA_A6   SEGA_D5 30 SEGA_D5
SEGA_A7 7 SEGA_A7   SEGA_D6 31 SEGA_D6
                          SEGA_D7 32 SEGA_D7
SEGA_A8 8 SEGA_A8   SEGA_D8 33 SEGA_D8
SEGA_A9 9 SEGA_A9   SEGA_D9 34 SEGA_D9
SEGA_A10 10 SEGA_A10   SEGA_D10 35 SEGA_D10
SEGA_A11 11 SEGA_A11   SEGA_D11 36 SEGA_D11
SEGA_A12 12 SEGA_A12   SEGA_D12 37 SEGA_D12
SEGA_A13 13 SEGA_A13   SEGA_D13 38 SEGA_D13
SEGA_A14 14 SEGA_A14   SEGA_D14 39 SEGA_D14
SEGA_A15 15 SEGA_A15   SEGA_D15 40 SEGA_D15

SEGA_A16 16 SEGA_A16   GND 41 GND
SEGA_A17 17 SEGA_A17
SEGA_A18 18 SEGA_A18   5V   42 +5V
SEGA_A19 19 SEGA_A19   GND 43 GND
SEGA_A20   20 SEGA_A20 3V   44 +3.3V
SEGA_A21   21 SEGA_A21
SEGA_A22   22 SEGA_A22
SEGA_A23   23 SEGA_A23 SEGA_OE   45 SEGA_OE
                         SEGA_WE_LB   46 SEGA_WE_LB
GND 24 GND   SEGA_WE_UB   47 SEGA_WE_UB
                         SEGA_CE   48 SEGA_CE
                         SEGA_REGISTER 49 SEGA_REGISTER
                         SEGA_AS   50 SEGA_AS
                         SEGA_RESET   51 SEGA_RESET
                         GND 52 GND

+5V   +3.3V   GND

PWR_FLAG PWR_FLAG PWR_FLAG

X1
Template for Kinetoscope
board header alignment

SRAM_WE 1 SRAM_WE   SRAM_DIS_0 3 SRAM_DIS_0
GND   2 GND   SRAM_DIS_1 4 SRAM_DIS_1
                          GND 5 GND
SRAM_W_A[0..18]

J2

◇ SRAM_D_IN[0..15]

SRAM_W_A0 6 SRAM_W_A0   SRAM_D_IN0 26 SRAM_D_IN0
SRAM_W_A1 7 SRAM_W_A1   SRAM_D_IN1 27 SRAM_D_IN1
SRAM_W_A2 8 SRAM_W_A2   SRAM_D_IN2 28 SRAM_D_IN2
SRAM_W_A3 9 SRAM_W_A3   SRAM_D_IN3 29 SRAM_D_IN3
SRAM_W_A4 10 SRAM_W_A4   SRAM_D_IN4 30 SRAM_D_IN4
SRAM_W_A5 11 SRAM_W_A5   SRAM_D_IN5 31 SRAM_D_IN5
SRAM_W_A6 12 SRAM_W_A6   SRAM_D_IN6 32 SRAM_D_IN6
SRAM_W_A7 13 SRAM_W_A7   SRAM_D_IN7 33 SRAM_D_IN7

SRAM_W_A8 14 SRAM_W_A8   SRAM_D_IN8 34 SRAM_D_IN8
SRAM_W_A9 15 SRAM_W_A9   SRAM_D_IN9 35 SRAM_D_IN9
SRAM_W_A10 16 SRAM_W_A10   SRAM_D_IN10 36 SRAM_D_IN10
SRAM_W_A11 17 SRAM_W_A11   SRAM_D_IN11 37 SRAM_D_IN11
SRAM_W_A12 18 SRAM_W_A12   SRAM_D_IN12 38 SRAM_D_IN12
SRAM_W_A13 19 SRAM_W_A13   SRAM_D_IN13 39 SRAM_D_IN13
SRAM_W_A14 20 SRAM_W_A14   SRAM_D_IN14 40 SRAM_D_IN14
SRAM_W_A15 21 SRAM_W_A15   SRAM_D_IN15 41 SRAM_D_IN15

SRAM_W_A16 22 SRAM_W_A16
SRAM_W_A17 23 SRAM_W_A17   GND 42 GND
SRAM_W_A18 24 SRAM_W_A18

GND 25 GND

These run on 3V, and output 3V logic, but tolerate 5V input.

U3
74LV161A

3	D0		Q0	14	A1
4	D1		Q1	13	A2
5	D2		Q2	12	A3
6	D3		Q3	11	A4
9	PE		TC	15	AR0
7	CEP			
10	CET			
2	CP			
1	MR			

CLK
CLR

+3.3V

U4
74LV161A

3	D0		Q0	14	A5
4	D1		Q1	13	A6
5	D2		Q2	12	A7
6	D3		Q3	11	A8
9	PE		TC	15	AR1
7	CEP			
10	CET			
2	CP			
1	MR			

CLK
CLR

There are not enough GPIO pins on the microcontroller
to write to SRAM in parallel, so we increment this counter
for the SRAM address.

+3.3V

U5
74LV161A

3	D0		Q0	14	A9
4	D1		Q1	13	A10
5	D2		Q2	12	A11
6	D3		Q3	11	A12
9	PE		TC	15	AR2
7	CEP			
10	CET			
2	CP			
1	MR			

CLK
CLR

+3.3V

U6
74LV161A

3	D0		Q0	14	A13
4	D1		Q1	13	A14
5	D2		Q2	12	A15
6	D3		Q3	11	A16
9	PE		TC	15	AR3
7	CEP			
10	CET			
2	CP			
1	MR			

CLK
CLR

+3.3V

U7
74LV161A

3	D0		Q0	14	A17
4	D1		Q1	13	A18
5	D2		Q2	12	A19
6	D3		Q3	11	
9	PE		TC	15	
7	CEP			
10	CET			
2	CP			
1	MR			

CLK
CLR

A[1..19]

+3.3V

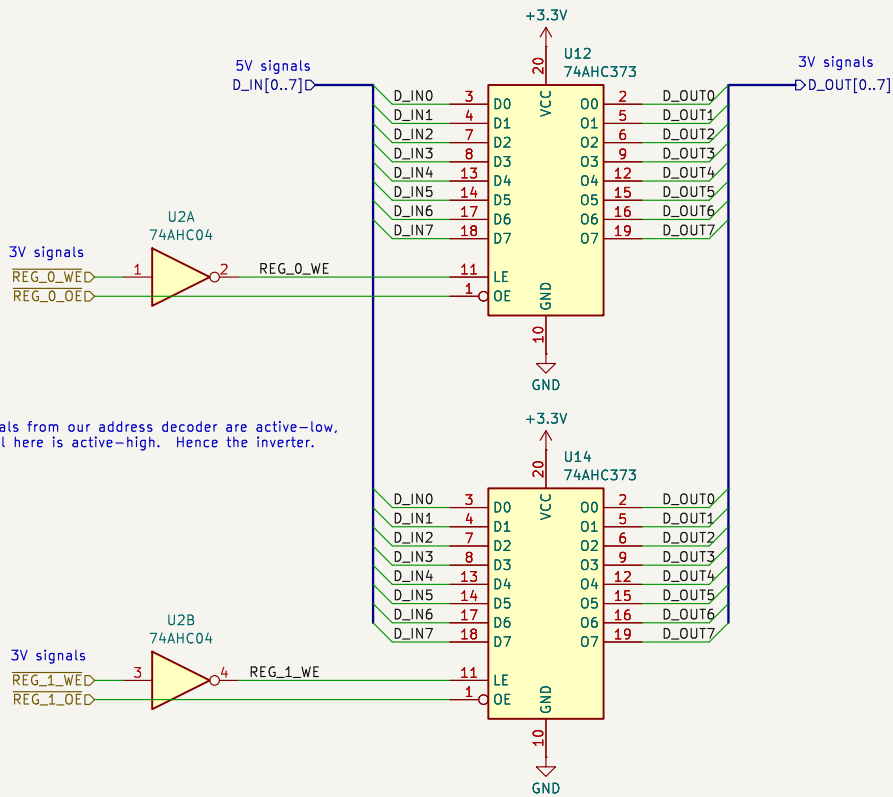| C3	C4	C5	C6	C7
22nF	22nF	22nF	22nF	22nF

GND

There are not enough GPIO pins on the microcontroller to write
to SRAM in parallel, so we fill this data register serially.

+3.3V

+3.3V
U8
74LV164A

MR	VCC	Q0	3	D0
DSA		Q1	4	D1
DSB		Q2	5	D2
CP		Q3	6	D3
		Q4	10	D4
		Q5	11	D5
		Q6	12	D6
	GND	Q7	13	D7

9 MR
1 DSA
2 DSB
8 CP

14 VCC
7 GND

GND

D_IN

CLK

D[0..15]

+3.3V
GND

C8
22nF

C9
22nF

+3.3V

+3.3V
U9
74LV164A

9 MR
1 DSA
2 DSB
8 CP

14 VCC
7 GND

Q0	3	D8
Q1	4	D9
Q2	5	D10
Q3	6	D11
Q4	10	D12
Q5	11	D13
Q6	12	D14
Q7	13	D15

GND

This is a pair of 8-bit registers that can be independently read
and written by the two processors (one reads, one writes).

The registers are assumed to written from the Sega's shared data bus
and read by the microcontroller. Each register has its own write-enable
and read-enable, but only one register can be read at a time, and only
one can be written at a time. It is safe to read from one register while
writing to another.

Everything runs on +3.3V, but must be compatible with 5V inputs. If using
a slightly different part number, please check input voltages in the data
sheet. The data outputs are 3.3V, which the Sega can read.

U12
74AHC373

U14
74AHC373

U2A
74AHC04

REG_0_WE

REG_0_WED
REG_0_OED

3V signals

5V signals
D_IN[0..7]

3V signals
D_OUT[0..7]

+3.3V

+3.3V

The write-enable signals from our address decoder are active-low,
but the latch signal here is active-high. Hence the inverter.

U2B
74AHC04

REG_1_WE

REG_1_WED
REG_1_OED

3V signals

+3.3V

C10
22nF

C11
22nF

C2
22nF

GND

U2G
74AHC04

These are unused gates.

U2C
74AHC04

U2E
74AHC04

U2D
74AHC04

U2F
74AHC04

GND

GND

GND

GND

These are synchronization tokens between the M68k and the microcontroller.

The tokens will be cleared on boot/reset.

The M68k will be able to set the command token ($\overline{\text{CMD\_SET}}$), indicating to
the microcontroller that a command is ready to be executed.

The microcontroller will be able to clear the command token ($\overline{\text{CMD\_CLR}}$),
indicating to the M68k that the command has been executed.

Command codes and arguments are passed through a separate register file.

The microcontroller can also flag an error ($\overline{\text{ERR\_SET}}$) to the M68k,
who can clear it ($\overline{\text{ERR\_CLR}}$) after noticing.

The M68k and microcontroller both will be able to read the current state
of either token through OUTPUT_5V, which goes to the shared data bus with
5V logic. 3.3V version of both tokens are output to the microcontroller
on dedicated lines.

U13A
74AHC74

GND
GND

U18A
74LVC2G125

Controlled by $\overline{\text{CMD\_OE}}$, outputs at 3.3V (OK) to M68k
on the shared data bus.

$\overline{\text{ERR\_OE}}$ and $\overline{\text{CMD\_OE}}$ must be mutually exclusive.

SHARED_OUTPUT

CMD_3V_ALWAYS_ON
Always on, outputs at 3.3V to the microcontroller.

+3.3V

U13C
74AHC74

C14
22nF

U18C
74LVC2G125

C16
22nF

GND

U13B
74AHC74

GND
GND

U18B
74LVC2G125

Controlled by $\overline{\text{ERR\_OE}}$, outputs at 3.3V (OK) to M68k
on the shared data bus.

$\overline{\text{ERR\_OE}}$ and $\overline{\text{CMD\_OE}}$ must be mutually exclusive.

SHARED_OUTPUT

ERR_3V_ALWAYS_ON
Always on, outputs at 3.3V to the microcontroller.

Sheet: /sync-token-internal/
File: sync-token-internal.kicad_sch

Title: Kinetoscope Sync Token

Size: A4   Date:

Rev:

KiCad E.D.A. 8.0.6   Id: 7/6