ISA bus

File: ECBbus.kicad_sch

Decoder

File: Decoder.kicad_sch

MMU four

File: MMU-4.kicad_sch

RTC

File: RTC.kicad_sch

ESP_IO

File: ESP_IO.kicad_sch

Power

File: Power.kicad_sch

CPU 6809

File: cpu-6809.kicad_sch

Spare

File: Spare.kicad_sch

ACIA

File: ACIA.kicad_sch

Memory Map

File: memory-map.kicad_sch

IO

File: IO.kicad_sch

Reset

File: Reset.kicad_sch

VIAS

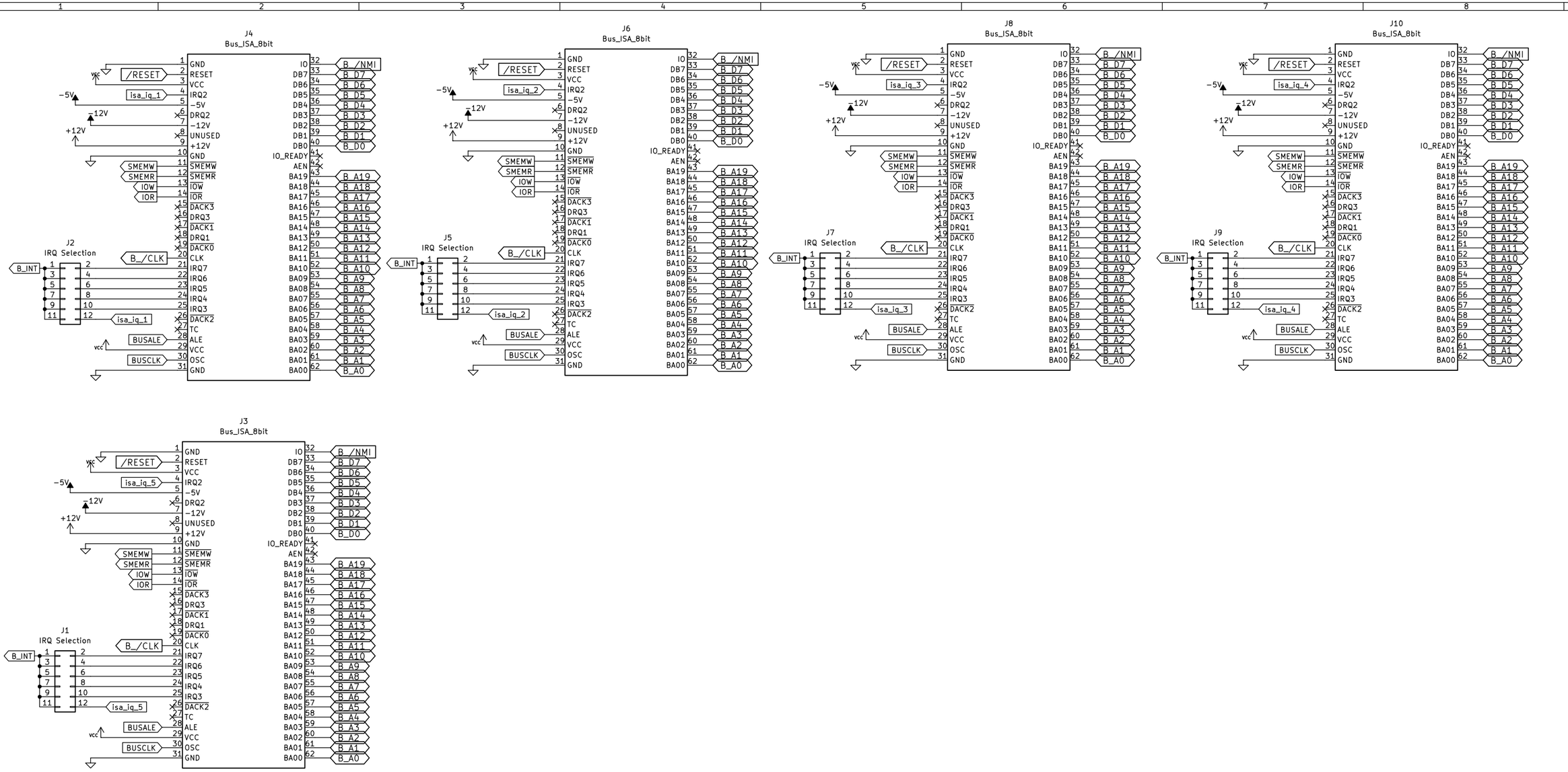
File: VIAS.kicad_sch

Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

Sheet: /
File: 6809PC-6U.kicad_sch

Title: 6809PC

Size: A	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 1/14



Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

Sheet: /ISA bus/
File: ECBbus.kicad_sch

Title: 6809PC

Size: USLedger	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 2/14

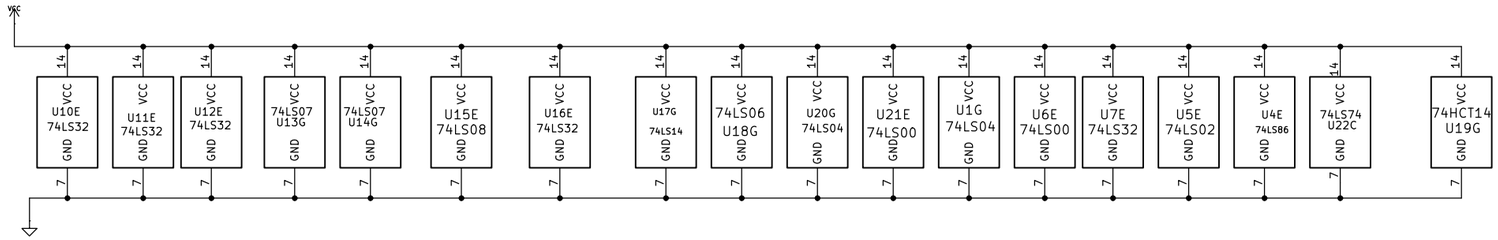
U1C
74LS04
U4A
74LS86
mA19 == RAM-A19
mA19
U4B
74LS86
mA19 != RAM-A19
U5B
74LS02
to DECODER sheet
SHADOW-IO
mA12
U1D
74LS04
U5C
74LS02
SHADOW-ROM
U5D
74LS02
U7A
74LS32
/VMA
from CPU sheet
U7B
74LS32
U6A
74LS00
/SHADOW
U3
COMP
74LS682
mA19
mA18
mA17
mA16
mA15
mA14
mA13
P=Q
P>Q
P4
SHADOW ADD
U9
29F040
mA0
mA1
mA2
mA3
mA4
mA5
mA6
mA7
mA8
mA9
mA10
mA11
mA12
mA13
mA14
mA15
mA16
mA17
mA18
mD0
mD1
mD2
mD3
mD4
mD5
mD6
mD7
PULLUP2
/RD
/WR
EPROM K1
EPROM/FLASH
FLASH
* for EPROM, A18 will satisfy the requirement of being equal to either Vih or Vil at all times.
U8
SRAM_512K
CE#
OE#
WE#
U2
74LS244
/MMU-IO-PREFIX-RD
Reads back the page number of the I/O space. The ROM boot space is IOpage+1.
Furthermore, the IOspace and ROMboot space overlay RAM, not ROM.
SHADOW ROM & I/O PREFIX READBACK (optional)
XM19
XM18
XM17
XM16
HI-ADDRESS BITS WHEN MMU IS DISABLED

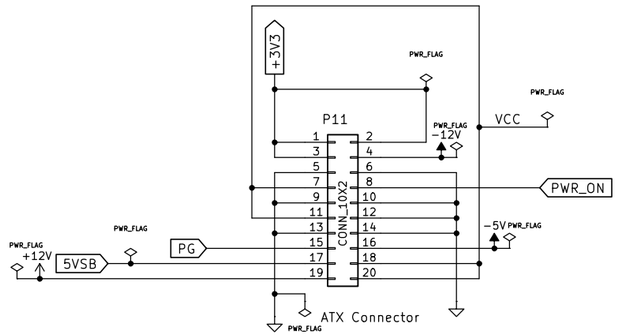
Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

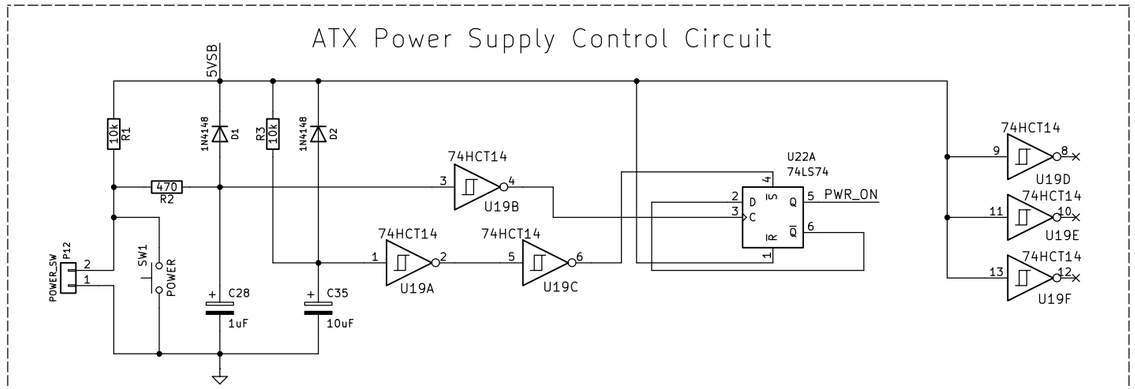
Sheet: /Memory Map/
File: memory-map.kicad_sch

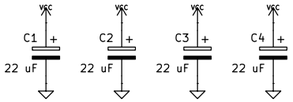
Title: 6809PC

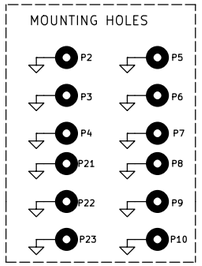
Size: B	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 4/14



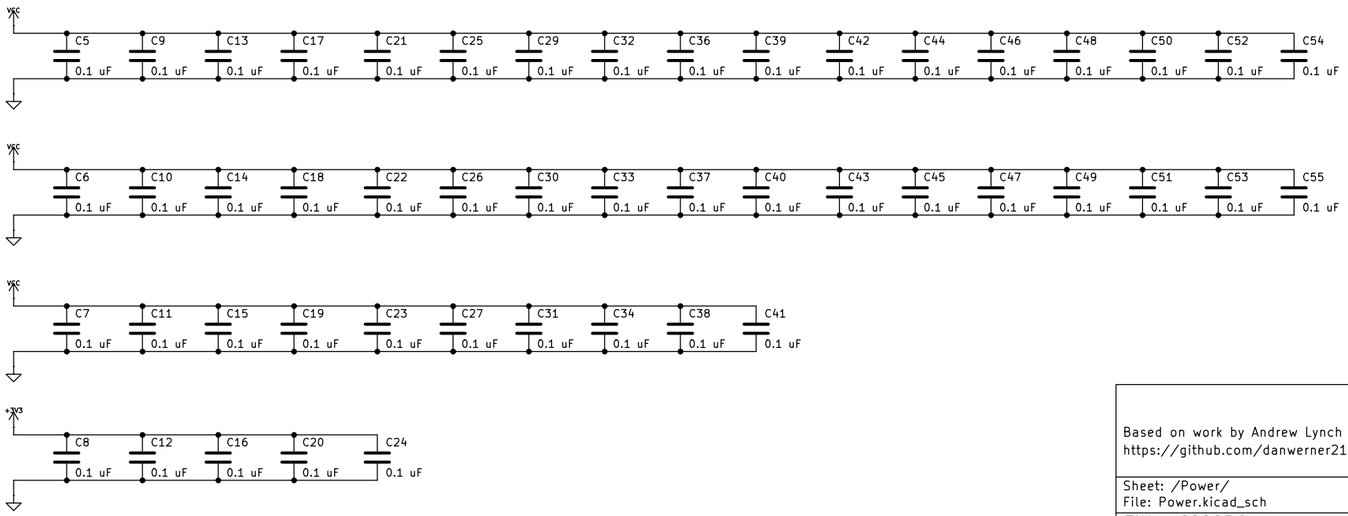








BYPASS CAPACITORS



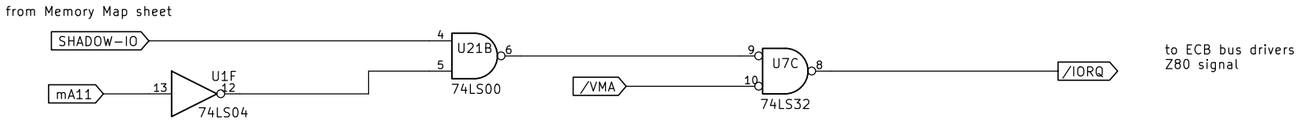
Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

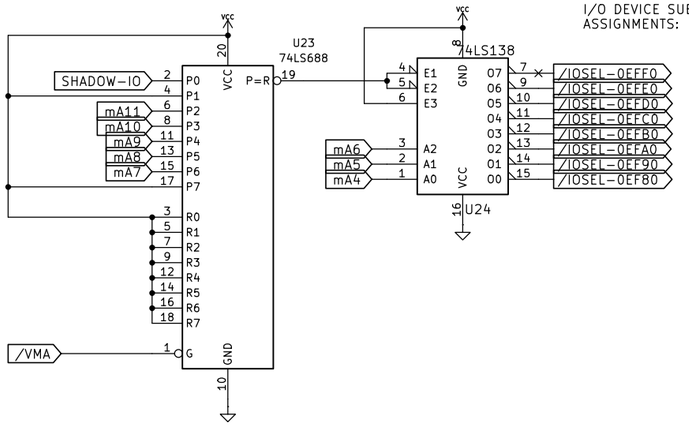
Sheet: /Power/
File: Power.kicad_sch

Title: 6809PC

Size: B	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 6/14

HIGHEST LEVEL I/O ADDRESS DECODE





I/O DEVICE SUBSYSTEM ADDRESS ASSIGNMENTS:

$EFF0 – $EFFF	available	
$EFE0 – $EFEF	MMU Registers (3)	
$EFD0 – $EFDF	MMU Task Map (16)	
$EFC0 – $EFCF	Video (3)	
$EFB0 – $EFBF	VIA0 (16)	
$EFA0 – $EFAF	VIA1 (16)	
$EF90 – $EF9F	RTC (16)	
$EF80 – $EF8F	ACIA (4)	

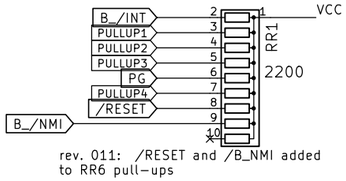
Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

Sheet: /Decoder/
File: Decoder.kicad_sch

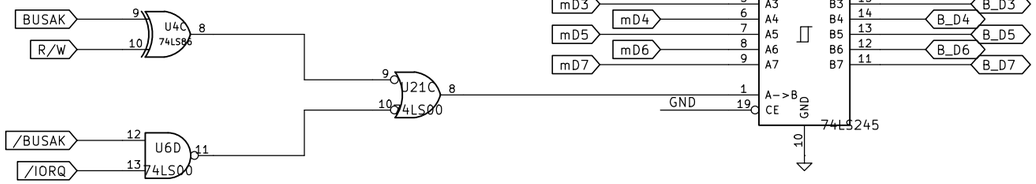
Title: 6809PC

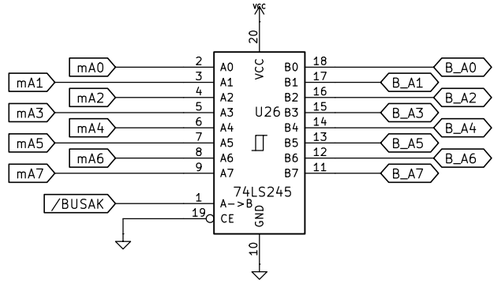
Size: B	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 7/14

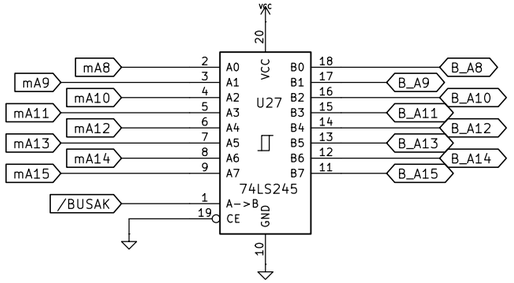
rev. 011: All signal equivalences are DELETED!!!

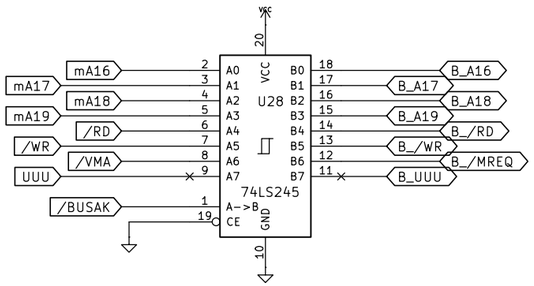


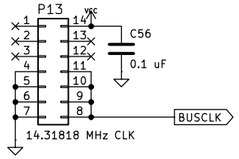
rev. 011: /RESET and /B_NMI added
to RR6 pull-ups

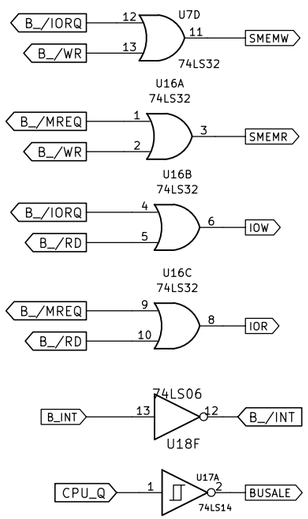












Bus Address Latch Enable. The address bus is latched on the rising edge of this signal.

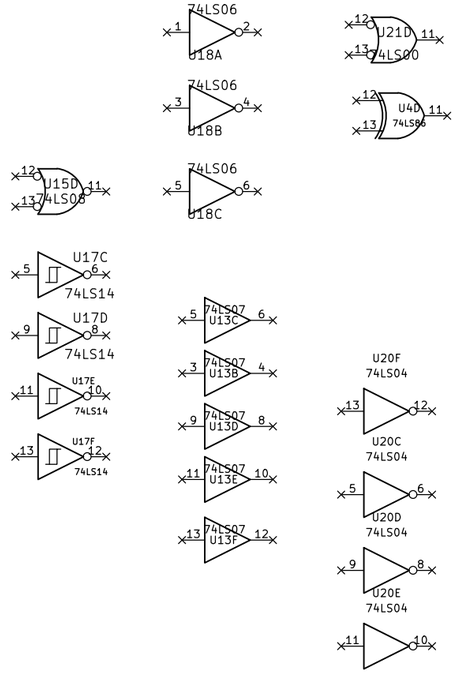
Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

Sheet: /IO/
File: IO.kicad_sch

Title: 6809PC

Size: B	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 8/14

SPARE

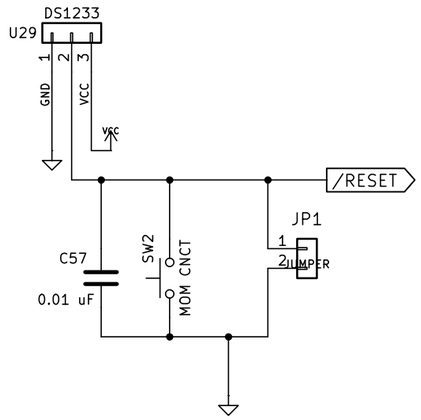


Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

Sheet: /Spare/
File: Spare.kicad_sch

Title: 6809PC

Size: B	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 9/14



Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

Sheet: /Reset/
File: Reset.kicad_sch

Title: 6809PC

Size: A	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 10/14

P14
UNIVERSAL OSCILLATOR
16.000MHz
VCC
C58
0.1 uF
U30
74LS163
8MHZ
4MHZ
2MHZ
1MHZ
P15
CPU CLOCK SELECT
8Mhz 1-2
4Mhz 3-4
2Mhz 5-6
1Mhz 7-8
CLK
GND
JP2
CPU_XTAL
U33
MC68B09
CPU_D0
CPU_D1
CPU_D2
CPU_D3
CPU_D4
CPU_D5
CPU_D6
CPU_D7
/RESET
/NMI
/IRQ
/FIRQ
/HALT
MRDY
LA0
LA1
LA2
LA3
LA4
LA5
LA6
LA7
LA8
LA9
LA10
LA11
LA12
LA13
LA14
LA15
CPU_R/W
CPU_BS
CPU_BA
CPU_E
CPU_Q
/HALT
MRDY
/NMI
/IRQ
/FIRQ
/MMU-DISABLE
/MMU-CLR-TASK-REG
CPU_BS
RR2
22K
E
U6B
74LS00
CPU_/RD
U1A
74LS04
CPU_R/W
W/R
U6C
CPU_E
CPU_/WR
U17B
74LS14
BUSAK
/BUSAK
BA has insufficient drive for the whole board
U1E
CPU_BA
CPU_BS
U21A
/INTA
CONN_3
/MMU-DISABLE
/MMU-CLR-TASK-REG
U5A
74LS02
CPU_E
CPU_Q
U32
74LS244
CPU_/RD
CPU_/WR
CPU_R/W
/IORQ
CPU_E
BUSAK
/VMA
similar to /MREQ
/RD
/WR
R/W
B_/IORQ
E
U31
74LS240
CLK
B_/NMI
B_/INT
B_/CLK
U18D
74LS06
/NMI
U18E
/IRQ
CPU_D7
CPU_D6
CPU_D5
CPU_D4
CPU_D3
CPU_D2
CPU_D1
CPU_D0
U32
74LS245
mD0
mD1
mD2
mD3
mD4
mD5
mD6
mD7
R/W
BUSAK

Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

Sheet: /CPU 6809/
File: cpu-6809.kicad_sch

Title: 6809PC

Size: B	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 11/14

Used during MMU setup/loading

Used during MMU mapping operation

U37 74LS157

rev. 011 -- change from /MMU-MAP-RW to MMU-ENABLE

U38 74LS374

/MMU-TASK-SETUP-WR

/MMU-MAP-RW

Optional TASK SETUP register

U40 SRAM-32K

SRAM - skinny 28-pin DIP 35ns 8K or 32K; e.g., any Cypress or JEDEC part.

U43 74LS257

U44 74LS257

MMU-ENABLE

U45 74LS244

U46 74LS244

Address Bus (tri-state) for all CPU's

ECB bus signal

U35 74LS174

Active TASK register

U39 74LS244

/MMU-TASK-RD

U36 74LS244

Read active TASK register and MMU-ENABLE status

U1B 74LS04

U15C 74LS08

/RESET

/MMU-CLR-TASK-REG

Select for programming the MMU map

U15A 74LS08

/VMA

Valid Memory Address

U10C 74LS32

U10D 74LS32

/IOSEL-0EFD0

/IOSEL-0EFE0

U42 74LS138

/MMU-IO-PREFIX-RD

/MMU-TASK-RD

/MMU-ENB-WR

/MMU-TASK-SETUP-WR

/MMU-TASK-WR

U41 74LS245

U22B 74LS74

/MMU-DISABLE

U15B 74LS08

/RESET

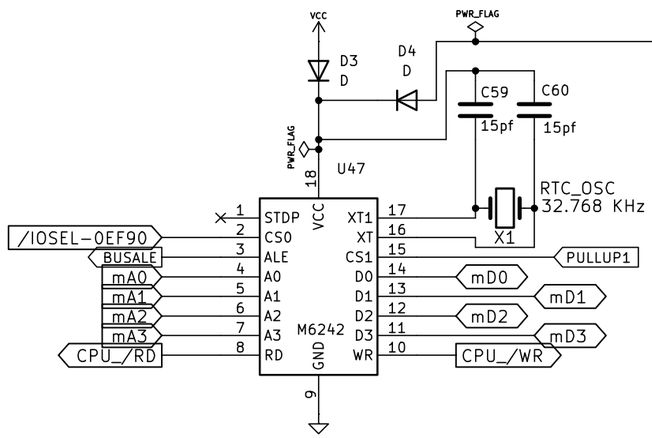
MMU enable/disable

Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

Sheet: /MMU four/
File: MMU-4.kicad_sch

Title: 6809PC

Size: B	Date: 2025-01-21	Rev: 001
KiCad E.D.A. 8.0.6		Id: 12/14



External Battery connector (2.5–3.6v)
or soldered Super Cap 0.1–1.0 Farad

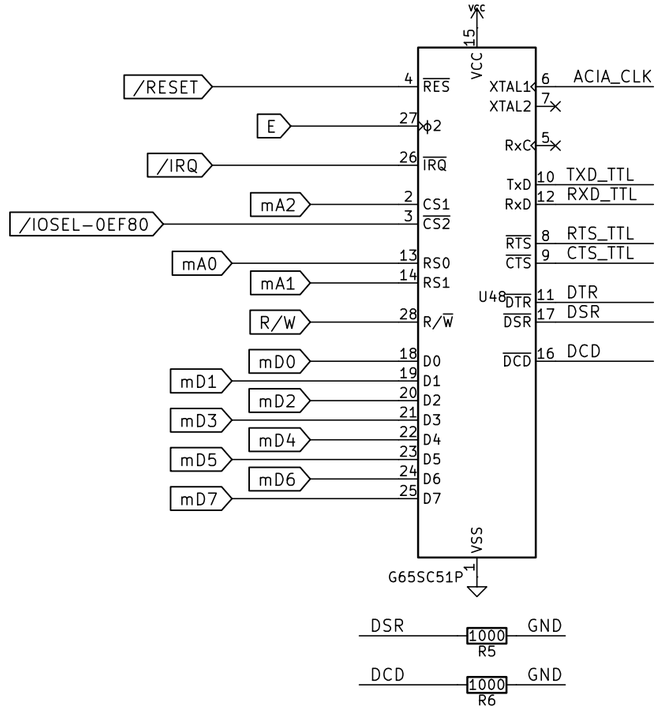
(e.g., Jameco 142949, 0.22F)

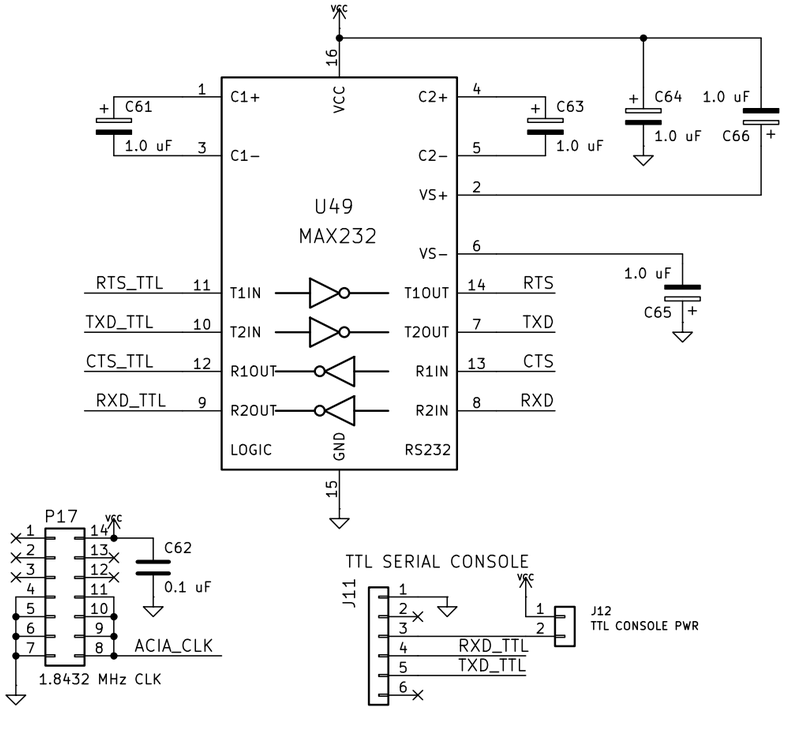
Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

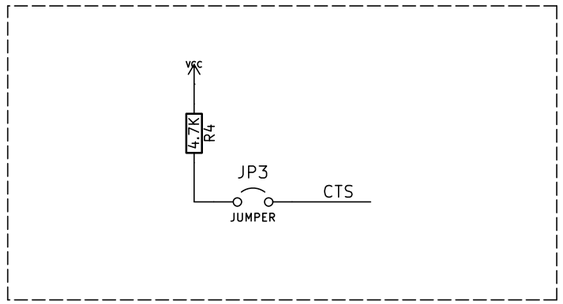
Sheet: /RTC/
File: RTC.kicad_sch

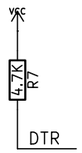
Title: 6809PC

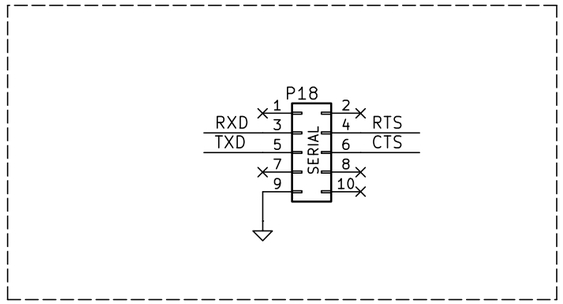
Size: A	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 13/14











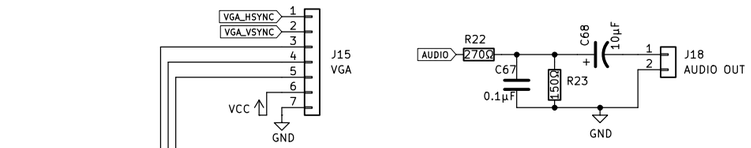
CTS is an inverted signal on the RS−232 port. So it is really /CTS. To assert the signal, it must be tied to SPACE, which is a + RS−232 voltage. (MARK, or true, is a − RS−232 voltage.)

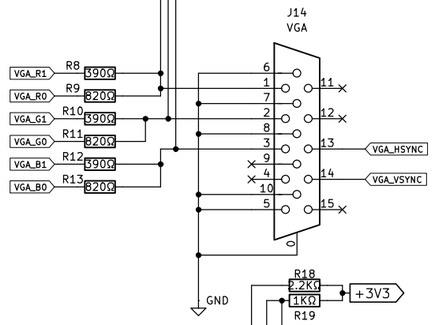
Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

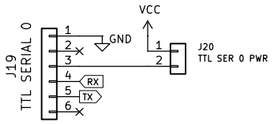
Sheet: /ACIA/
File: ACIA.kicad_sch

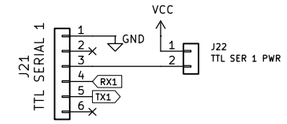
Title: 6809PC

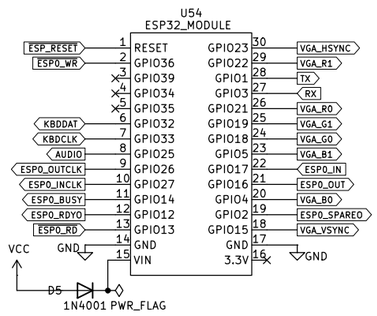
Size: A	Date: 2025−01−21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 14/14

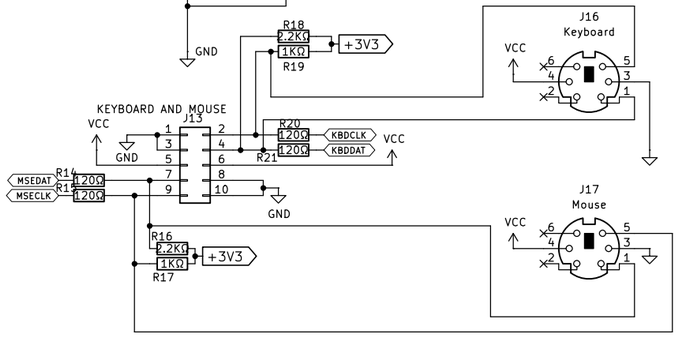


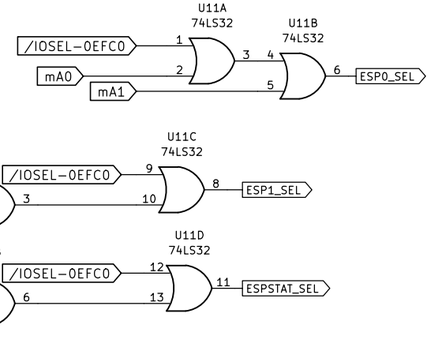


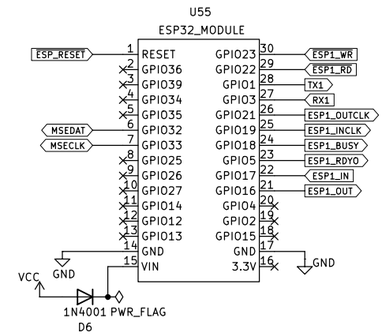


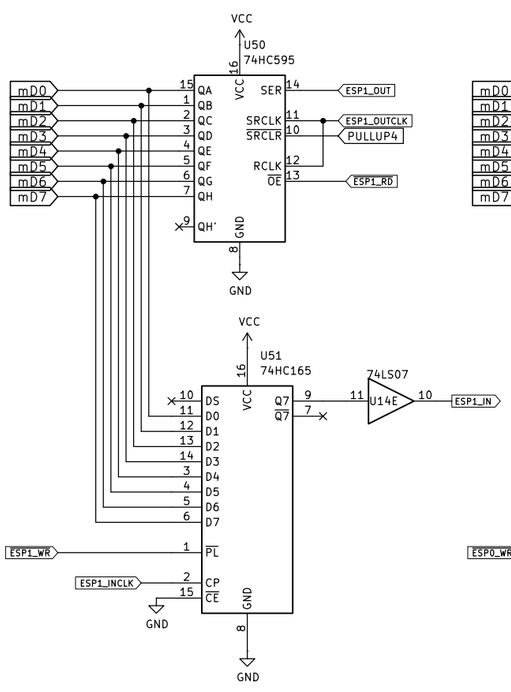


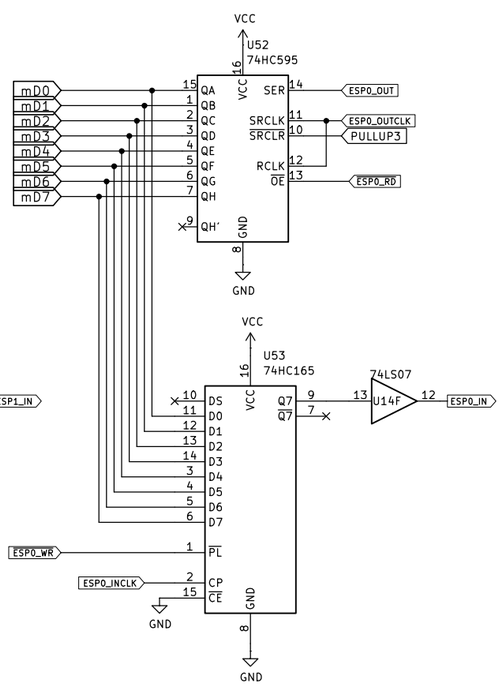


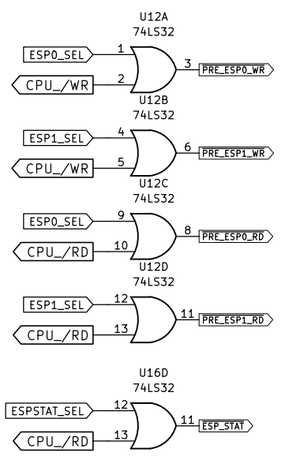


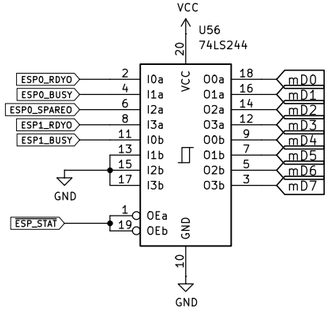


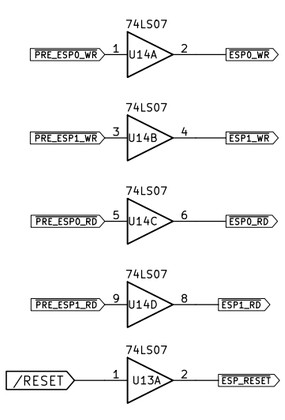


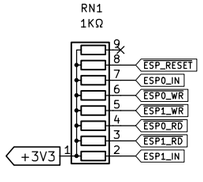










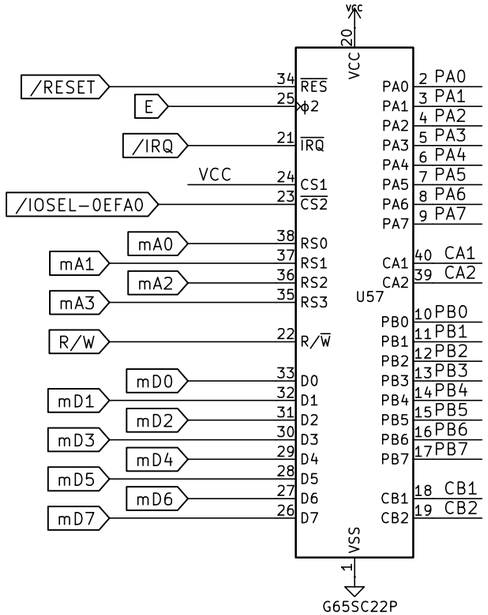


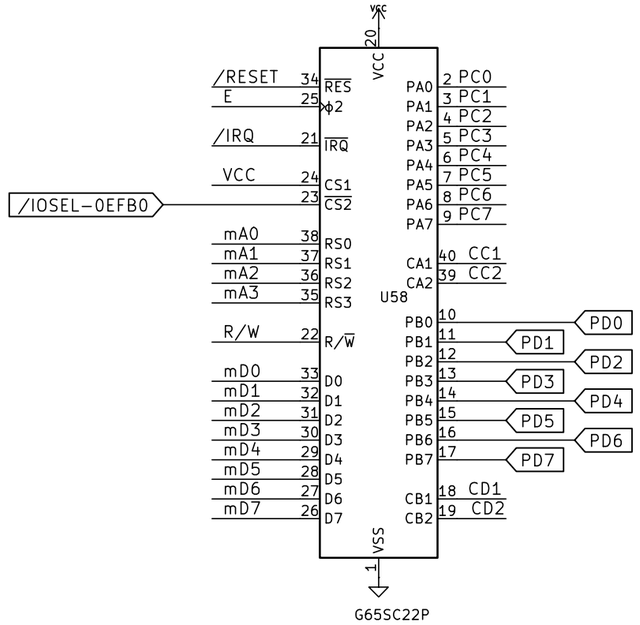
Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

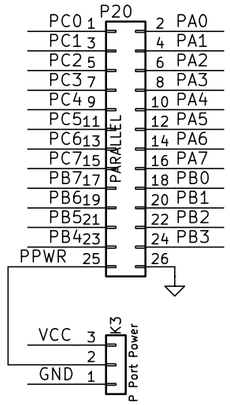
Sheet: /ESP_IO/
File: ESP_IO.kicad_sch

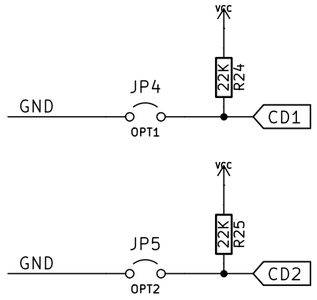
Title: 6809PC

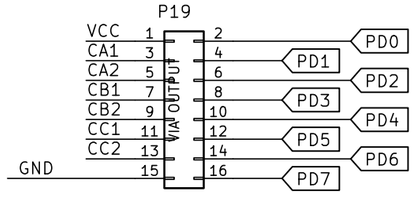
Size: USLedger	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 14/14











Based on work by Andrew Lynch and John Coffman
https://github.com/danwerner21/6809PC

Sheet: /VIAS/
File: VIAS.kicad_sch

Title: 6809PC

Size: A	Date: 2025-01-21	**Rev: 001**
KiCad E.D.A. 8.0.6		Id: 15/14