CPU

COPROCESSOR

DRAM CTRLR

32 MB DRAM

FLASH

SRAM

WAIT STATE

DRAM CONFIGURATION JUMPERS		
WAITIN	ADDW	WAIT STATES
ON	ON	4
ON	OFF	3
OFF	ON	3
OFF	OFF	2

GRYPHON
mc68030 retrocomputer

CPU/Memory | Sheet 1 of 4 | Rev 1.1

Paul D. Fincato/Dave Mehaffy 2012-2015 | 15 March 2015 | B

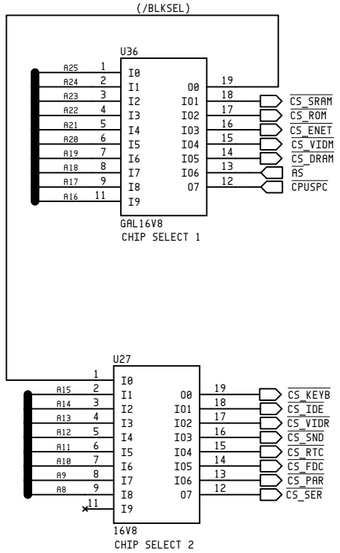
Home of retro computing - http://n8vem-sbc.pbworks.com/

CHIP SELECT 1

```
GAL16V8
CSEL1.2

A25  A24       A23  A22        A21      A20      A19     A18     A17    GND
A16  /CPUSPC   AS   /CSDRAM   /CSVRAM  /CSENET  /CSROM  /CSRAM  /BLKSEL VCC


BLKSEL = CPUSPC * /AS  * /A25 * /A24 * /A23 * /A22 * /A21 *  A20 * /A19 * /A18 * /A17 * /A16
CSROM  = CPUSPC * /AS  * /A25 * /A24 * /A23 * /A22 * /A21 * /A20 * /A19
CSRAM  = CPUSPC * /AS  * /A25 * /A24 * /A23 * /A22 * /A21 * /A20 *  A19
CSENET = CPUSPC * /AS  * /A25 * /A24 * /A23 * /A22 * /A21 *  A20 * /A19 * /A18 * /A17 *  A16
CSVRAM = CPUSPC * /AS  * /A25 * /A24 * /A23 * /A22 *  A21
CSDRAM = CPUSPC * /AS  *  A25


DESCRIPTION:

INPUTS
A25-A16 Address

OUTPUTS
Chip selects
BlockSelect to next address decoder

Rev 1.1 - Updated/corrected equations by Yoda
```

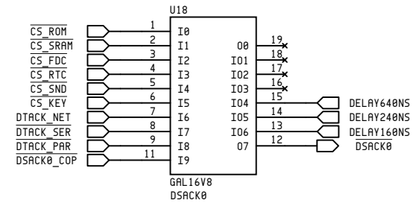
CHIP SELECT 2

```
GAL16V8
CSEL2.1

/BLKSEL A15    A14    A13    A12    A11    A10     A9      A8       GND
NC      /CSSER  /CSPAR /CSFDC /CSRTC /CSSND /CSVIDR /CSIDE  /CSKEYB  VCC


CSSER  =  BLKSEL * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 * /A9 * /A8
CSPAR  =  BLKSEL * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 * /A9 *  A8
CSFDC  =  BLKSEL * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 *  A9 * /A8
CSRTC  =  BLKSEL * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 *  A9 *  A8
CSSND  =  BLKSEL * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 * /A8
CSVIDR =  BLKSEL * /A15 * /A14 * /A13 * /A12 *  A11 * /A10 * /A9 * /A8
CSIDE  =  BLKSEL * /A15 * /A14 * /A13 * /A12 *  A11 * /A10 * /A9 *  A8
CSKEYB =  BLKSEL * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 *  A8
```



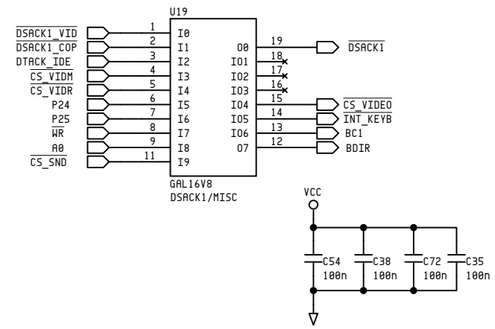
DSACK0 LOGIC

```
GAL16V8
DTACK0.1

/ROM   /RAM  /FDC  /RTC  /SND  /KEY  DSNET  /DSER /DPAR  GND
/DSCOP /DS0  DLA   DLB  DLC    /DA   /DB    NC    NC     VCC

DA  =  ROM * DLA
     + RAM * DLA
     + FDC * DLB
     + RTC * DLC
     + SND * DLC
     + KEY * DLC
     + /DSNET

DB  = DSER
    + DPAR
    + DSCOP

DS0 = DA + DB
```



DSACK1 / MISC DECODE

```
GAL16V8
DTACK1.1

/DVID  /DCOP DIDE  /CSVM  /CSVR  P24  P25  /WR  A0    GND
/CSSND /BDIR /BC1  /INTKB /CSVID NC   NC   NC   /DS1  VCC

DS1 =  DVID + DCOP + /DIDE ; DSACK1

CSVID = CSVM + CSVR        ; VIDEO

INTKB = P24 + P25          ; KEYB INT

BDIR = CSSND * WR          ; SOUND CS
BC1 =  CSSND * A0
```



MEMORY MAP

```
DEVICE  BASE ADDRESS            SIZE (BYTES)
        HEX                     DEC             HEX
-------------------------------------------------------------
ROM     00000000-0007FFFF       524288          00080000
SRAM    00080000-000FFFFF       524288          00080000
SERPORT 00100000-001000FF       256             00000100
PARPORT 00100100-001001FF       256             00000100
FDC     00100200-001002FF       256             00000100
RTC     00100300-001003FF       256             00000100
SOUND   00100400-001004FF       256             00000100
KEYBD   00100500-00100SFF       256             00000100
---     00100600-001009FF
VIDREG* 00100A00-00100AFF       256             00000100
IDE*    00100B00-00100BFF       256             00000100
---     00100C00-0010FFFF
ETHNET  00110000-0011FFFF       65536           00010000
---     00120000-001FFFFF
VIDRAM* 00200000-003FFFFF       2097152         00200000
---     00400000-01FFFFFF
DRAM#   02000000-03FFFFFF       33554432        02000000
---     04000000-FFFFFFFF

*16 bit device / #32 bit device

IDE is base address for drive 0
Drive 1 is base + 128
```

BYTE DECODE

```
GAL16V8
BYDEC.2

A0 A1   SIZ0 SIZ1 RW   CSROM CSRAM CSENET CSVRAM GND
NC /UUD /UMD /LMD /LLD CSDRAM CSVIDR NC /IOEN VCC

UUD = RW + /A0 * /A1

UMD = RW
    + A0  * /A1
    + /A1 * /SIZ0
    + /A1 *  SIZ1

LMD = RW
    + /A0 * A1
    + /A1 * /SIZ0 * /SIZ1
    + /A1 *  SIZ0 *  SIZ1
    + /A1 *  A0   * /SIZ0

LLD = RW
    + A0    *  A1
    + A0    *  SIZ0 * SIZ1
    + /SIZ0 * /SIZ1
    + A1    *  SIZ1

IOEN = CSROM * CSRAM * CSENET * CSVRAM * CSDRAM * CSVIDR

DESCRIPTION:

Byte select logic on 32 bit bus
MC68030 User's manual page section 12-13
Figure 12-7

Rev 1.1 - Updated/corrected equations by Yoda
```

DRAM INTERFACE

```
GAL16V8
DRMGLU.2

BCLK /CS    /AS /WR /DTACK  NC  /ADDW  CLK    NC     GND
/OE  /STERM /DC NC /DB    /DA   MLCLK   /ENCAS  /AREQ  VCC

AREQ  = AS    * CS *  CLK
      + AREQ * CS * /CLK

ENCAS = AREQ * CS * /DC
      + AREQ * CS * /CLK

DC =  AS * CS * DB * /CLK
    + AS * CS * DC * CLK

STERM = AS * CS * DA * /DB * /CLK * ADDW
      + AS * CS * DTACK * /DB * /CLK * /ADDW
      + /STERM * CLK

MLCLK= CS * AS * WR

DA.R  = AREQ * CS * DTACK * /DB * ADDW

DB.R  = AREQ * CS * DTACK * DA * /DB * ADDW
      + AREQ * CS * DTACK * /DB * /ADDW


DESCRIPTION:

DP8422V DRAM Controller Interface logic
National Application Note AN-537
"Interfacing the DP8420A/21A/22A to the 68030 Microprocessor"
Page 3

PDF -
converted from National PLAN format
removed EXST from equations (External STERM)

Rev 1.1 - Updated/corrected equations by Yoda
```

COPROCESSOR DECODE

```
GAL16V8
COPRO.1

CLK  AS  FC2  FC1 FC0 A19 A18 A17 A16  GND
A15 /CS /CLKD A14 A13 NC  NC  NC  CPU  VCC

CS =  FC2 *  FC1 * FC0
   * /A19 * /A18 * A17 * /A16
   * /A15 * /A14 * A13
   * /CLK

   +  FC2 *  FC1 * FC0
   * /A19 * /A18 * A17 * /A16
   * /A15 * /A14 * A13
   * /AS

   +  FC2 *  FC1 * FC0
   * /A19 * /A18 * A17 * /A16
   * /A15 * /A14 * A13
   * /CLKD

CLKD = CLK

; PDF - added for peripheral chipselect1 logic
CPU = FC2 * FC1 * FC0


DESCRIPTION:

MC68882 chip select logic
MC68030 User's manual section 12-8
Figure 12-4
```

NOTES:

8422V PROGRAMMING

TI DP8422V Datasheet Ref. Page 9 section 3.2.2

Chip select access programming. Reset causes /DISRFSH to assert and FF to assert /ML. The first write of CPU on boot MUST provide 8422 programming data with program word on address bus, asserting /CS_DRAM. Upon write completion FF will negate /ML and write cycle terminates via normal /STERM. 8422 will be ready for normal operation.

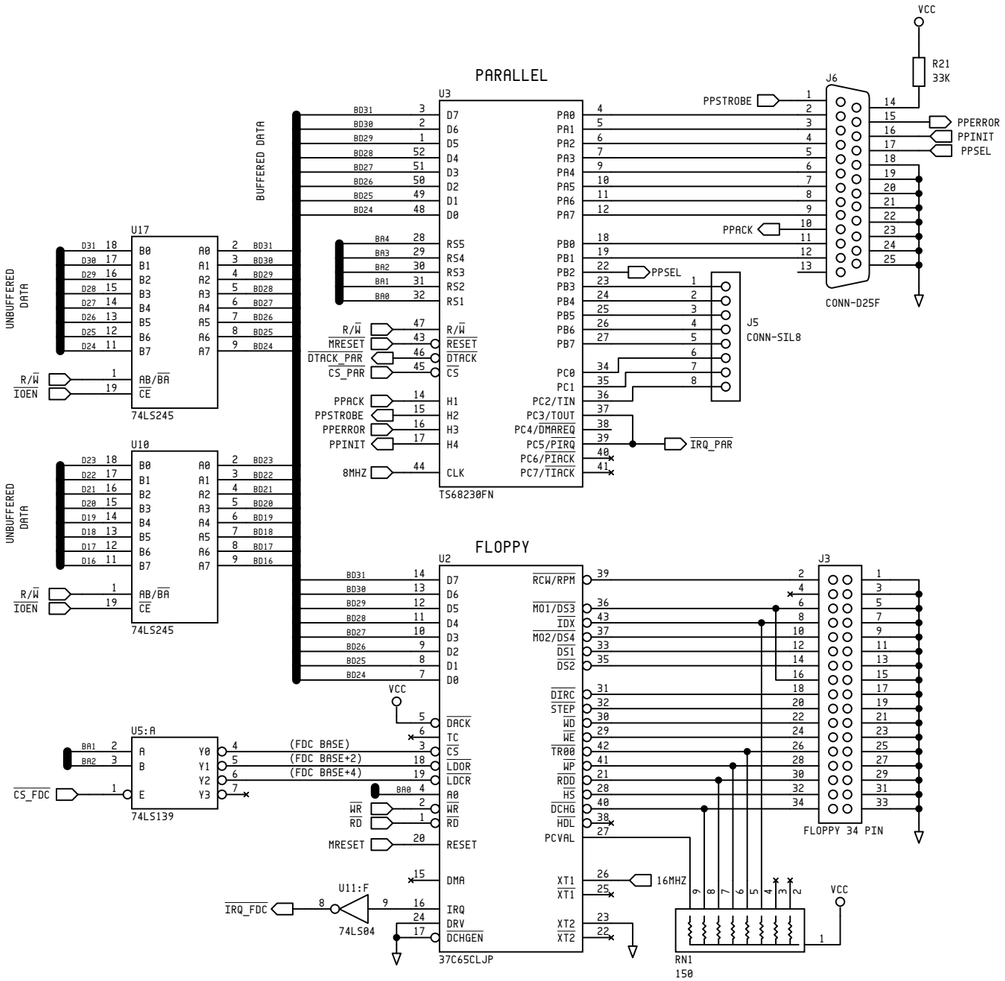
GAL LOGIC COMPILER

All GAL equations can be compiled with GALASM

https://github.com/daveho/GALasm

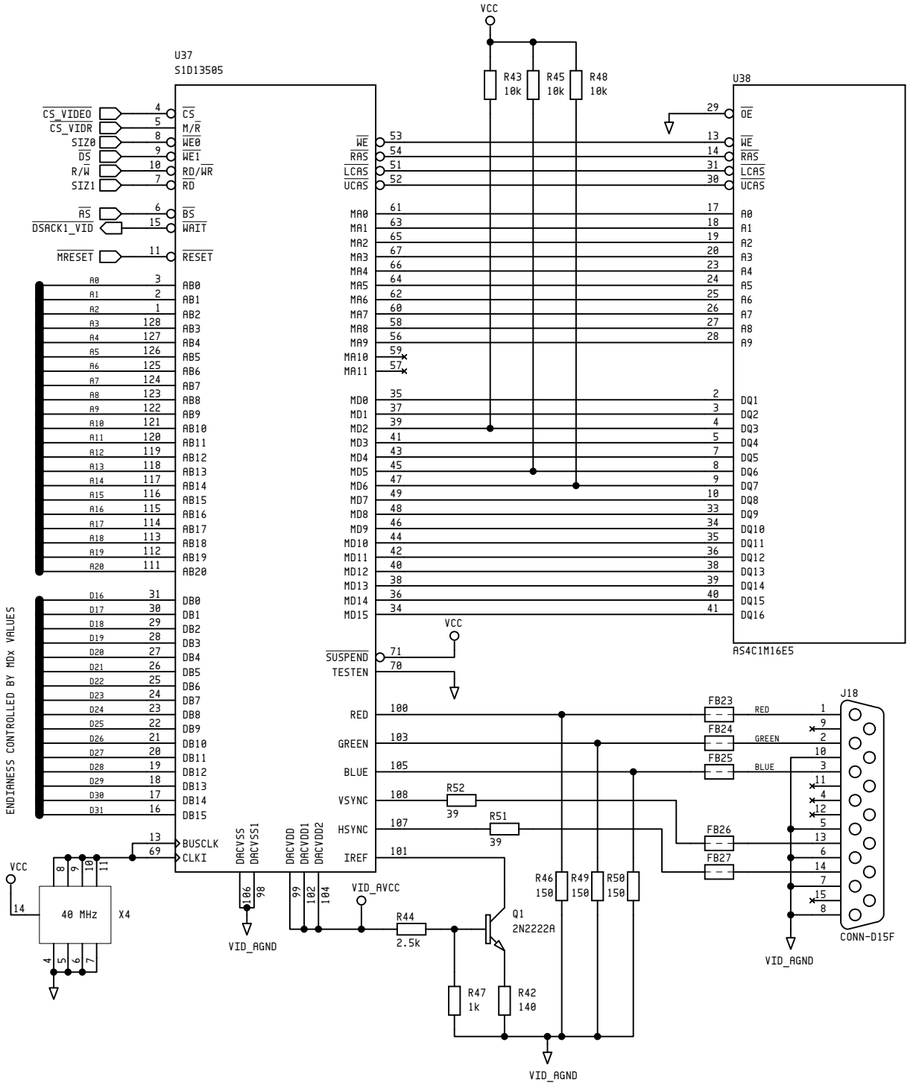
GRYPHON
mc68030 retrocomputer

Glue Logic	Sheet 2 of 4	Rev 1.1
Paul D. Fincato/Dave Meharfy 2012-2015	15 March 2015	B
Home of retro computing - http://n8vem-sbc.pbworks.com/		

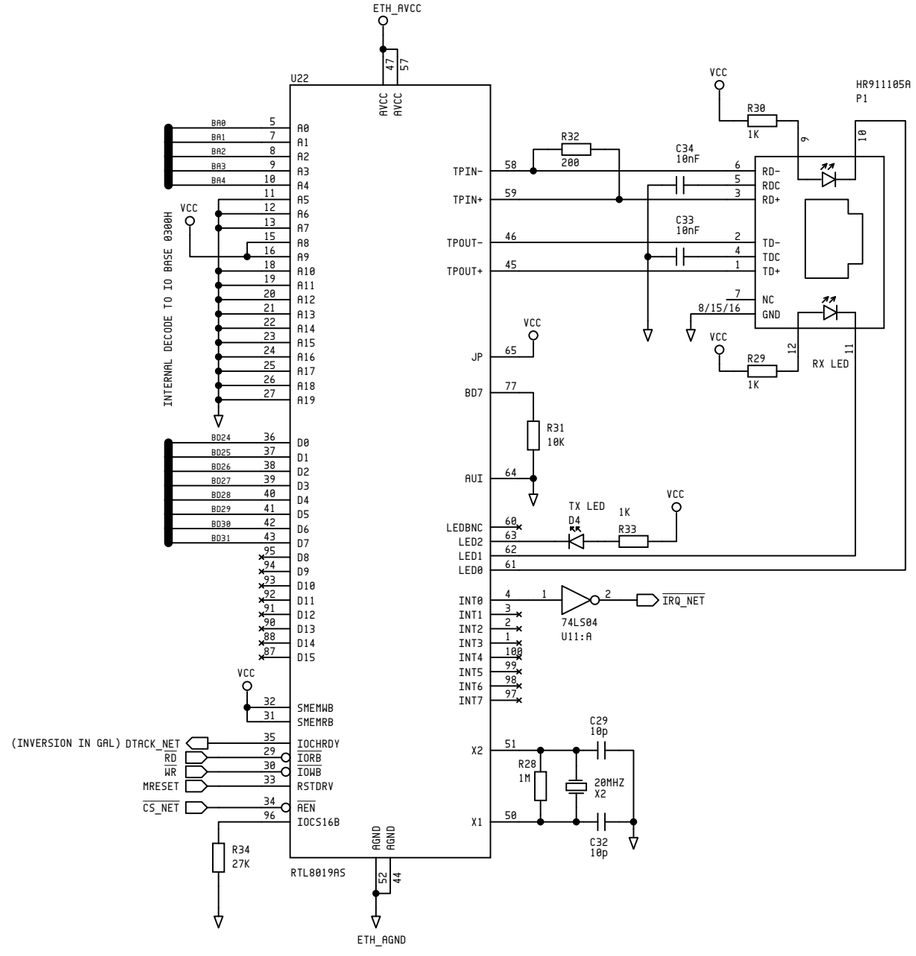


GRYPHON mc68030 retrocomputer		
Peripherals	Sheet 3 of 4	Rev 1.1
Paul D. Fincato/Dave Mehaffy 2012-2015	15 March 2015	B
Home of retro computing - http://n8vem-sbc.pbworks.com/		

VIDEO



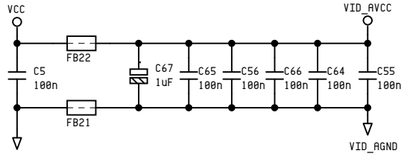
ETHERNET

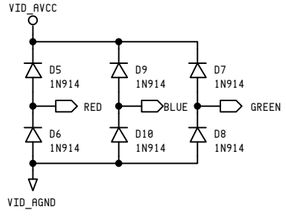


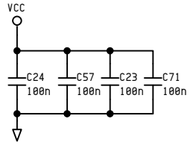
S1D13505 CONFIGURATION

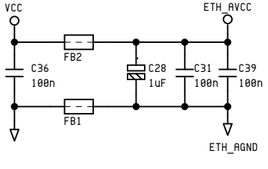
	16 BIT HOST	MC68K BUS2	MC68K BUS2	MC68K BUS2	BIG ENDIAN	WAIT# - WAIT HIGH	SYMMETRIC 1MB	SYMMETRIC 1MB		SUSPEND# INPUT	GPO POLARITY	PRIMARY BUS INTF	BUSCLK 1:1 CLK
MD	0	1	2	3	4	5	6	7	8	9	10	11	12
VALUE	0	0	(1)	0	0	(1)	(1)	0	X	0	0	0	0

0 = INTERNAL PULLDOWN









GRYPHON
mc68030 retrocomputer

Video/Ethernet	Sheet 4 of 4	Rev 1.1
Paul D. Fincato/Dave Mehaffy 2012-2015	15 March 2015	B
Home of retro computing - http://n8vem-sbc.pbworks.com/		