Clock/Calendar Information

The clock/calendar is based on the OKI MSM6242RS Direct Bus Connected-Type Real Time Clock Chip.

The A2000 features a real time clock with a perpetual calendar which is capable of reading and writing "YEAR", "MONTH", "DAY", "WEEK", "HOUR", "MINUTE" and "SECOND". This time clock is a peripheral IC, connected directly by means of a bus. It is standard on the A2000, and can be added as an option to the A500 (included in the A501 Memory Expander).

An interface between the time clock and a microcomputer uses 4 of data bus lines, 4 address bus lines, 3 control bus lines and 2 chip select pins, and performs time setting, reading and other operations.

The clock function covers second, minute, hour, day, month, year and day of week. In addition, other functions such as selection of a 24-hour time and a 12-hour time system, automatic adjustment of leap year in the Christian Era and 30-second correction by means of soft, periodical interruption (or periodical wave-form output) and stop/start of time counting.

The clock-calendar is a CMOS device, so there is low power consumption.

A crystal used is capable of 32.768 KHz for a consideration over time counting during battery backup.

When installed, the clock is located in memory at $DC0000.

Clock Warning
The addresses used by the real time clock chip access the custom chip registers without the memory expansion/real time clock module. When probing to test for the existence of the clock, care must be taken to avoid unintentional changes to the custom chip register. The test used by the setclock utility references an address that maps to either the seconds register or a static read-only chip register, then checks to see if the clock "ticks."

Note: C912 can be used as a slow/fast control to tune the clock to best effect.

REGISTER TABLE

Address	A_3	A_2	A_1	A_0	Number of Register	Data				Count value	Description
						D_3	D_2	D_1	D_0		
0	0	0	0	0	S_1	S_8	S_4	S_2	S_1	0 ~ 9	1-second digit register
1	0	0	0	1	S_{10}	*	S_{40}	S_{20}	S_{10}	0 ~ 5	10-second digit register
2	0	0	1	0	MI_1	mi_8	mi_4	mi_2	mi_1	0 ~ 9	1-minute digit register
3	0	0	1	1	MI_{10}	*	mi_{40}	mi_{20}	mi_{10}	0 ~ 5	10-minute digit register
4	0	1	0	0	H_1	h_8	h_4	h_2	h_1	0 ~ 9	1-hour digit register
5	0	1	0	1	H_{10}	*	PM/ AM	h_{20}	h_{10}	0 ~ 2 or 0 to 1	PM/AM, 10-hour digit register
6	0	1	1	0	D_1	d_8	d_4	d_2	d_1	0 ~ 9	1-day digit register
7	0	1	1	1	D_{10}	*	*	d_{20}	d_{10}	0 ~ 3	10-day digit register
8	1	0	0	0	MO_1	mo_8	mo_4	mo_2	mo_1	0 ~ 9	1-month digit register
9	1	0	0	1	MO_{10}	*	*	*	mo_{10}	0 ~ 1	10-month digit register
A	1	0	1	0	Y_1	y_8	y_4	y_2	y_1	0 ~ 9	1-year digit register
B	1	0	1	1	Y_{10}	y_{80}	y_{40}	y_{20}	y_{10}	0 ~ 9	10-year digit register
C	1	1	0	0	W	*	w	w	w	0 ~ 6	Week register
D	1	1	0	1	C_D	30 sec. ADJ	IRQ FLAG	BUSY	HOLD	—	Control Register D
E	1	1	1	0	C_E	t_1	t_0	ITRPT /STND	HASK	—	Control Register E
F	1	1	1	1	C_F	TEST	24/ 12	STOP	REST	—	Control Register F

- 0–low level 1–high level
- REST–RESET
- ITRPT/STND–INTERRUPT/STANDARD

Note 1: You have the option to write data into the bit*. However, this data is treated as 0 internally. In addition, the bit* is always read as 0.

Note 2: You can write 1 into the IRQ FLAG bit and 0 or 1 into the BUSY bit. They are not executed, but can be read.

Note 3: It is possible to read and write all bits other than bit* and BUSY bit. However, only 0 can be written into IRQ FLAG.

More information on the clock/calendar may be found in the OKI MSM6242RS Direct Bus Connected-Type Real Time Clock app. notes.

Power Budgets

B2000 POWER BUDGET

A2000/B2000 POWER SUPPLY:

All of the specifications herein are suggested. When it comes right down to it, the machine is being powered by a well-defined supply, the specifications of which will follow. If you're careful not to exceed the suggested load for any port, you'll be able to fully load every port. However, some of the internal ports can draw more than the suggested amount; for example, an 8 megabyte expansion memory card for the 100 pin bus may draw more than the suggested 2.5 Amps at +5VDC. The connector is capable of supplying more without damage, but the extra current must be carefully worked into the system power budget. Any hardware add-on device that draws more than the suggested amount must state this clearly. External ports typically have a true maximum available, not a suggested; budgeting should apply to internal items only.

VOLTAGE	SYSTEMWIDE LIMIT	DESCRIPTION
+ 5 VDC	20.0 Amps	Main +5 Voltage supply
+ 5 USER	0.5 Amps	Protected +5 for externals
− 5 VDC	0.3 Amps	Negative 5 Volt supply
+12 VDC	8.0 Amps	Main +12 Voltage supply
+12 USER	---------	Protected +12 for externals, derived from main +12
−12 VDC	0.3 Amps	Main −12 Voltage supply
−12 USER	---------	Protected −12 for externals, derived from main −12

CONSUMPTION:

Everybody wants power. Here's what can be taken, based on your particular setup; you get what's left over:

MAIN SYSTEM:	+5VDC	−5VDC	+5USER	+12VDC	−12VDC
Motherboard	2.5A	---	---	50mA	50mA
Internal 3½" Floppy [1]	250mA	---	---	350mA	---
Internal 5¼" Floppy [2]	500mA	---	---	500mA	---
Internal 3½" Hard Disk [2]	750mA	---	---	1.0A	---
Internal 5¼" Hard Disk [2]	1.0A	---	---	1.5A	---
EXTERNAL PORTS:					
Video Port	---	10mA	100mA	100mA	---
Floppy Port [1]	250mA	---	---	350mA	---
Parallel Port [3]	---	---	10mA	---	---
Serial Port	---	---	---	25mA	25mA
Keyboard Port [4]	250mA	---	---	---	---
Mouse Port	---	---	50mA	---	---
INTERNAL SLOTS:					
CoProcessor Slot [6]	2.0A	40mA	---	40mA	35mA
Expansion Slot [6]	2.0A	40mA	---	40mA	35mA
Extra PC Bus Slots [7]	0.5A	10mA	---	40mA	15mA
Video Slot [8]	1.0A	40mA	---	40mA	---

NOTES:

[1] Expected typical consumption. This is very device dependent; consult the manufacturer's specification for particular floppy disks. The starting current is expected to be around 400mA for +12V.
[2] Expected typical consumption. This is very device dependent; consult the manufacturer's specification for particular disks. Starting current on the +12V supply for most disk drives can be as much as twice the operating current.
[3] 47 Ohm Series Resistor limits current.
[4] Expected typical consumption. Current from this port is limited.
[5] Each port.
[6] Each slot. The physical connection can handle 4 Amps; if a 4 Amp device is used in one slot, other slots cannot supply 2.5 Amps each, of course; this requires a total system power budget to be constructed.
[7] Shared PC expansion slots should be considered part of the 100 pin connector that they share. If the 100 pin connector is unused, the power suggested for that connector can be used instead of the PC bus. The connectors, like all expansion connectors, are capable of delivering 4 Amps if proper whole-system budgeting is done. The specification here is for both of the non-overlapping PC slots taken together.
[8] Like expansion slots the video slot is capable of supplying 4. If a 4 Amp device is used, it must be worked into the total system power budget.

A500 Power Budget

PARALLEL PORT:

10mA from pin 14 (+5V)
(47Ω series resistor to prevent damage if printer grounds this line)

SERIAL PORT:

20mA from pin 9 (+12V)
20mA from pin 10 (−12V)
(47Ω series resistor to limit current)

VIDEO PORT:

100mA from pin 23 (+5V)	No
100mA from pin 22 (+12V)	current
10mA from pin 21 (−12V)	limiting

JOYSTICK PORTS (TOTAL)

50mA from pins 7 (+5V)
(4.7Ω current limit)

EXPANSION PORT:

300mA from pins 5 and 6 (+5V)	No
50mA from pin 10 (+12V)	current
10mA from pin 8 (−12V)	limiting

A2000 PAL Equations

```
PAL20L8'                                          PAL DESIGN SPECIFICATION
PART NO.: 380 XXX-01  DESCRPT.:PALEN     REV.2    FRANK ULLMANN   03-09-86
MEM- AND DTACK-DECODER FOR A2500 MAINBOARD (U26) ASSY 380...
COMMODORE BSW          !! PRELIMENARY !!

A23 A22 A21 A20 A19 A18 PRW AS DBR OVL OVR GND
C1 C3 VPA MYRAME CLKE RGAE RE DTACK BLS ROME XRDY VCC


IF (OVR) /VPA = /AS*A23*/A22*A21                    ; PERIPHERAL ACCESS
                                                    ; $A00000-BFFFFF

/MYRAME  = /AS*DTACK*A23*A22*A21*OVR*/C1*C3         ; $E00000-FFFFFF
         + /AS*DTACK*/A23*/A22*/A21*OVR*OVL*/C1*C3  ; $000000-1FFFFF IF
                                                    ; OVL=H, OVR=H !
         + /AS*DTACK*A23*A22*/A21*A20*A19*/A18*OVR*/C1*C3   ; $D80000-DBFFFF
         + /MYRAME*/C1
         + /MYRAME*/C3

/RE      = DBR*/AS*DTACK*/A23*/A22*/A21*OVR*/OVL*   ; $000000-1FFFFF IF
           /C1*C3                                   ; OVL=L, OVR=H !
         + /RE*/C1
         + /RE*/C3

IF (OVR) /DTACK =
           /AS*/A23*/A22*A21*XRDY                   ; $200000-3FFFFF EXP RAM
         + /AS*/A23*A22*XRDY                        ; $400000-7FFFFF  "   "
         + /AS*A23*/A22*/A21*XRDY                   ; $800000-9FFFFF  "   "
         + /MYRAME*XRDY*/C3                         ; $000000-1FFFFF OVL=H
                                                    ; AND $E00000-FFFFFF
         + /RE*/C3                                  ; $000000-1FFFFF OVL=L
         + /RGAE*/C3                                ; $C00000-D7FFFF
                                                    ; AND $DC0000-DF0000
         + /DTACK*/AS*XRDY

/RGAE    = DBR*/AS*DTACK*A23*A22*/A21*A20*/A19*   OVR*/C1*C3;$D00000-$D7FFFF
         + DBR*/AS*DTACK*A23*A22*/A21*A20*A19*A18*OVR*/C1*C3;$DC0000-$DFFFFF
         + DBR*/AS*DTACK*A23*A22*/A21*/A20*       OVR*/C1*C3;$C00000-$CFFFFF
         + /RGAE*/C1
         + /RGAE*/C3

/BLS     = /AS*DTACK*/A23*/A22*/A21*OVR*/OVL*/C1*C3 ; $000000-1FFFFF OVL=L
         + /AS*DTACK*A23*A22*/A21*OVR*/C1*C3        ; $C00000-DFFFFF
         + /BLS*/C1
         + /BLS*/C3
```

```
/ROME    = /AS*A23*A22*A21*A20*A19*OVR*PRW          ; $F80000-FFFFFF
         + /AS*A23*A22*A21*/A20*/A19*OVR*PRW        ; $E00000-E7FFFF
         + /AS*/A23*/A22*/A21*/A20*/A19*OVR*OVL*PRW ; $000000-07FFFF
         + /AS*/A23*/A22*/A21*A20*A19*OVL*OVR*PRW   ; $180000-1FFFFF

/CLKE    = /AS*A23*A22*/A21*A20*A19*/A18*OVR        ; $D80000-DBFFFF
```

DESCRIPTION

THE CLOCK IS NOW TILED THROUGHOUT THE SPACE $D80000-DBFFFF. IF MORE PRECISE SELECTION TO $D80000 $D8FFFF IS REALLY NEEDED, THEN THIS MUST BE DONE EXTERNALLY USING THE /CS INPUT ON THE CLOCK CHIP.

DTACK FOR THE CLOCK IS HANDLED IN THE MYRAME EQUATION BECAUSE THE DTACK EQUATION ALREADY HAS 7 OR TERMS! THIS MEANS THAT CLOCK ACESSES WILL BE SYNCHRONIZED TO VIDEO CYCLES, BUT THIS SHOULD CREATE NO MAJOR PROBLEMS.

THE IMPLEMENTATION OF ROME/MYRAME MATCHES THE A1000 - IT MIGHT BE DESIRABLE TO HAVE THE ADDRESS RANGE IN MYRAME MATCH THE ROM SELECT ADDRESS RANGE...

MYRAME OUTPUT IS NOT USED EXTERNALLY - ONLY INTERNAL USAGE!!!

```
PAL16L8                                          PAL DESIGN SPECIFICATION
PART NO.: 380 XXX-01  DESCRPT.:PALCAS    REV.1   FRANK ULLMANN   08-29-86
RAM/ROM-DECODER FOR A2500 MAINBOARD REV.2 (U27) ASSY 380...
COMMODORE BSW        !! PRELIMENARY !!

ARW C1 C3 PRW UDS LDS RE RGAE CLKE GND
DBR CLKR RRW LCEN UCEN CDR CDW DAE CLKW  VCC


/CDR     = /RE*PRW*C1                            ; ENABLE RAM READ
         + /RGAE*PRW*C1                          ; BUFFER
         + /CDR*/LDS
         + /CDR*/UDS

/CDW     = /RE*/PRW                              ; ENABLE RAM WRITE
         + /RGAE*/PRW                            ; BUFFER
         + /CDW*C1

/UCEN    = /DAE*RE*C1                            ; GENERATE CAS
         + DAE*/RE*/UDS*C1                       ; SIGNALS
         + /UCEN*C1

/LCEN    = /DAE*RE*C1                            ; GENERATE CAS
         + DAE*/RE*/LDS*C1                       ; SIGNALS
         + /LCEN*C1

/RRW     = /RE*/PRW                              ; /WE FOR DRAMS CPU
         + /DAE*/ARW*/C1                         ; OR AGNUS ACCESS
         + /RRW*/DAE

/DAE     = /DBR*/C1*C3                           ; CHIP RAM ADDR ENABLE
         + /DAE*/C1
         + /DAE*/C3

/CLKR    = /CLKE*/LDS*PRW                        ; CLOCK READ

/CLKW    = /CLKE*/LDS*/PRW                       ; CLOCK WRITE
```

DESCRIPTION

PAL20L8 PAL DESIGN SPECIFICATION

PART NO.: 380715-2 DESCRPT.: PAL BUFFER CONTROL REV.2 HEINZ ULLRICH 06-18-87

PAL BUFFER CONTROL FOR A2500 (U5) FOR PRODUCTION-PCB

COMMODORE BSW

SLV1 SLV2 SLV3 SLV4 SLV5 OVR RD BAS RESET A23 A22 GND

A21 A20 D2P A19 BERR OWN DS NCOLLIS PROC DBOE ASQ VCC

```
/NCOLLIS  =      SLV1*SLV2*SLV3*SLV4*SLV5
          + PROC*     SLV2*SLV3*SLV4*SLV5
          + PROC*SLV1*      SLV3*SLV4*SLV5
          + PROC*SLV1*SLV2*      SLV4*SLV5
          + PROC*SLV1*SLV2*SLV3*      SLV5
          + PROC*SLV1*SLV2*SLV3*SLV4

/PROC     = /BAS*/A23*/A22*/A21*          RESET*OVR
          + /BAS* A23*/A22* A21*          RESET*OVR
          + /BAS* A23* A22*/A21*          RESET*OVR
          + /BAS* A23* A22* A21*/A20*/A19*RESET*OVR
          + /BAS* A23* A22* A21* A20* A19*RESET*OVR

/D2P      = OWN*/SLV1*RD            ; DOWNSTREAM READS UPSTREAM SLAVE
          + OWN*/SLV2*RD            ;          -"-
          + OWN*/SLV3*RD            ;          -"-
          + OWN*/SLV4*RD            ;          -"-
          + OWN*/SLV5*RD            ;          -"-
          + /OWN*SLV1*SLV2*SLV3*SLV4*SLV5*/RD ; UPSTREAM WRITES DOWNSTREAM SL

/DBOE     = /BAS*/RD    *BERR*OWN      ; CPU READS FROM SLAVE
          + /D5*RD*/ASQ*BERR*OWN       ; CPU WRITES TO SLAVE
          + /BAS*/RD    *BERR*/OWN*SLV1*SLV2*SLV3*SLV4*SLV5; DMA READS CPU RAM
          + /DS*RD*/ASQ*BERR*/OWN*SLV1*SLV2*SLV3*SLV4*SLV5; DMA WRITES TO CPU RAM

IF (RESET*NCOLLIS) /BERR = RESET
```

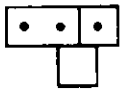
DESCRIPTION

```
PAL16R6                                                PAL DESIGN SPECIFICATION
PART NO.:          DESCRPT.:PAL ARBITRATE A2500 REV.1 FRANK ULLMANN 05-22-86
PAL ARBITRATE FOR A2500 (U  ) FOR PRODUCTION-PCB
COMMODORE BSW

7M BAS RES BGIN BR5 BR4 BR3 BR2 BR1 GND
NC BASD BGOLD BG5 BG4 BG3 BG2 BG1 BR VCC


/BG1    = RES*/BGIN*BGOLD*/BR1          ; GENERATE BG1
        + RES*/BGIN*/BG1                ; HOLD UNTIL /BG

/BG2    = RES*/BGIN*BGOLD*/BR2*BR1
        + RES*/BGIN*/BG2

/BG3    = RES*/BGIN*BGOLD*/BR3*BR1*BR2
        + RES*/BGIN*/BG3

/BG4    = RES*/BGIN*BGOLD*/BR4*BR1*BR2*BR3
        + RES*/BGIN*/BG4

/BG5    = RES*/BGIN*BGOLD*/BR5*BR1*BR2*BR3*BR4
        + RES*/BGIN*/BG5

/BGOLD  = /BGIN                         ; STORE OLD STATE OF BG

/BR     = RES*/BR5                      ; BR IS REQUEST TO 68K
        + RES*/BR1
        + RES*/BR2
        + RES*/BR3
        + RES*/BR4

/BASD   = BAS


DESCRIPTION

BG1 IS HIGHEST PRIORITY
```

List of B2000 Motherboard Jumpers

J101

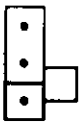
3 2 1



This jumper determines the high-order address bit for Fat Agnus. In its normal position shown the high-order bit is A23; in its other position, this bit is A19. The current Fat Agnus chip requires the A23 signal for proper management of the memory at $C00000. Future Fat Agnus chips may map things differently.

J200

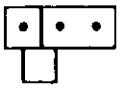
1
2
3



This jumper is used to set the light-pen port number. In the normal position shown, the light pen input will be the FIRE input of mouse/joystick port 1, as with the A500. With the jumper in the other position, the light pen input will be the FIRE input of mouse/joystick port 0, which is the scheme used on the A1000 machine.

J300

1 2 3



This jumper determines the time base used for the 50/60Hz CIA timer chip. In the normal position, the 50/60Hz TICK clock, based on AC line frequency, is used as a time base. In the alternate position, the vertical sync pulse from the video section is used. The system will not operate properly without one of these clocks.

J301

X X

This jumper is closed to add a second internal floppy drive, open to leave the second floppy out of the main unit box.

J500

X-X

This jumper is used to enable the 512K of RAM at $C00000. It is normally closed; opening it will disable this extra RAM.