# CHG

## **C**T60 **H**ARDWARE **G**UIDE

Rev 5.2 – October 2000 – February 2004


**Try to use this file on screen and not print it on paper ! Remember that paper is produced with trees !!**

## SUMMARY

1 - FUNCTIONAL BLOCK DIAGRAM

2 - ADDRESSES & REGISTERS

3 - 060 BUS SLOT

4 - HARDWARE EMULATION

5 - THERMAL SENSOR

6 - DIMM EEPROM

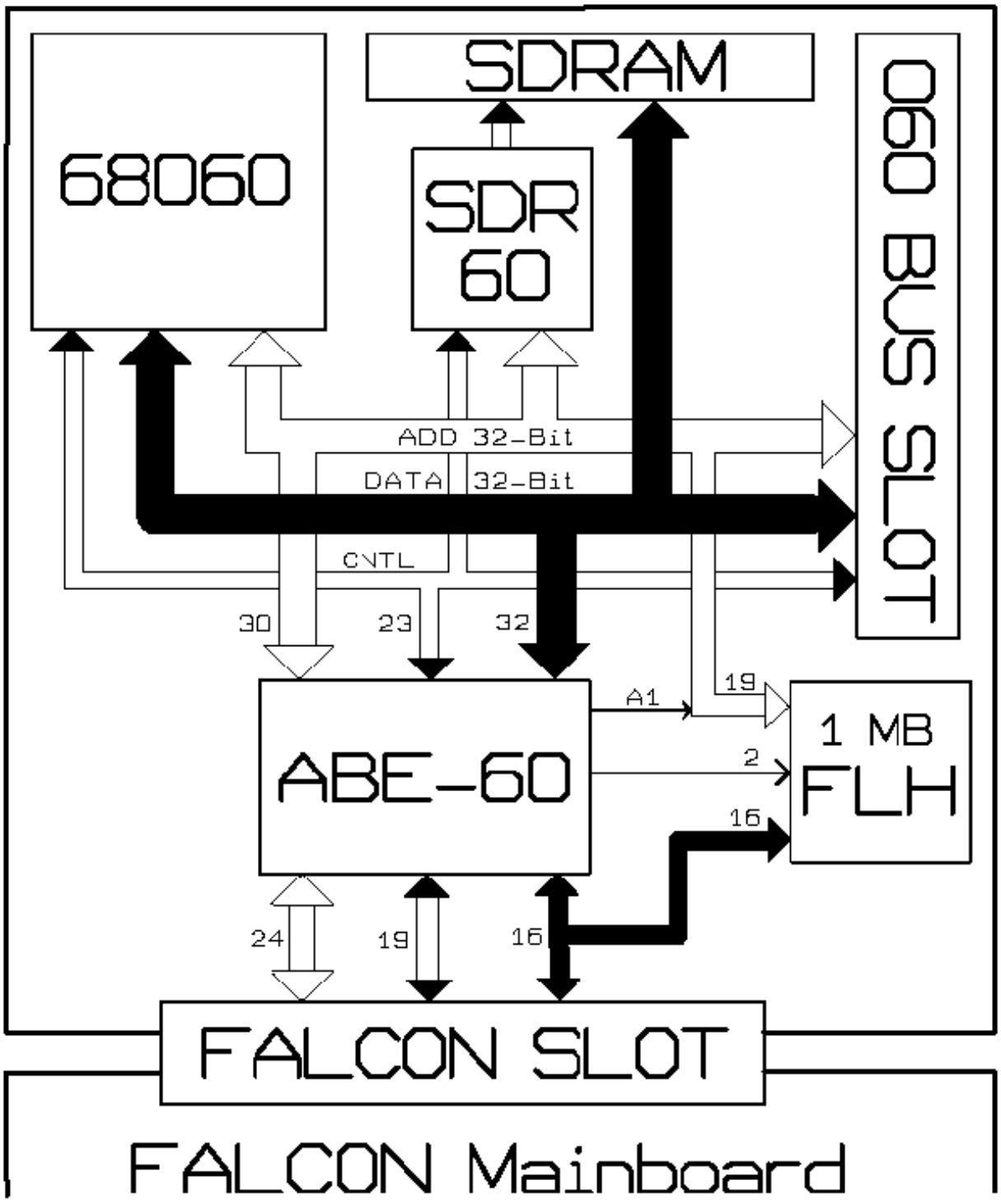
7 - DIMM SDRAM for CT60

8 - BURST with SDRAM

9 - INTERRUPTS

10 - CHIPSET pinouts

## FUNCTIONAL BLOCK DIAGRAM




CT60 FUNCTIONAL BLOCK DIAGRAM

# ADDRESSES & REGISTERS

## 68030 VIEW 24-Bit MAP

| | | | |
|---|---|---|---|
| $xx000000 | $xxDFFFFF | 14 MB | ST-RAM |
| $xxE00000 | $xxEFFFFF | **1 MB** | **TOS 4.0x ROM - BOOT** |
| $xxE00000 | $xxEFFFFF | **1 MB** | **CT60 FLASH – CPU SPACE #3** |
| $xxF00000 | $xxF0FFFF | 64 KB | I/O IDE |
| $xxF10000 | $xxF9FFFF | 576 KB | F030 BUS SLOT |
| $xxFA0000 | $xxFBFFFF | 128 KB | CARTRIDGE SLOT |
| $xxFC0000 | $xxFEFFFF | 192 KB | Unused |
| $xxFF0000 | $xxFFFFFF | 64 KB | I/O |

## 68060 VIEW 32-Bit MAP

| | | | | |
|---|---|---|---|---|
| $00000000 | $00DFFFFF | 14 MB | ST-RAM | CACHE - NO BURST |
| **$00E00000** | **$00EFFFFF** | **1 MB** | **CT60 FLASH** | CACHE - NO BURST |
| $00F00000 | $00F0FFFF | 64 KB | I/O IDE | NO CACHE - NO BURST |
| $00F10000 | $00F9FFFF | 576 KB | F030 BUS SLOT | NO CACHE - NO BURST |
| $00FA0000 | $00FBFFFF | 128 KB | CARTRIDGE SLOT | NO CACHE - NO BURST |
| $00FC0000 | $00FEFFFF | 192 KB | Unused | NO CACHE - NO BURST |
| $00FF0000 | $00FFFFFF | 64 KB | I/O | NO CACHE - NO BURST |
| **$01000000** | **$04FFFFFF** | **64 MB** | **SDRAM (TT-RAM)** | CACHE - BURST |
| **$01000000** | **$08FFFFFF** | **128 MB** | **SDRAM (TT-RAM)** | CACHE - BURST |
| **$01000000** | **$10FFFFFF** | **256 MB** | **SDRAM (TT-RAM)** | CACHE - BURST |
| **$01000000** | **$20FFFFFF** | **512 MB** | **SDRAM (TT-RAM)** | CACHE - BURST |
| $21000000 | $3FFFFFFF | 496 MB | Reserved | CACHE - BURST |
| **$40000000** | **$7FFFFFFF** | **1024 MB** | **060 BUS SLOT** | CACHE - BURST |
| **$80000000** | **$BFFFFFFF** | **1024 MB** | **060 BUS SLOT** | NO CACHE - NO BURST |
| $C0000000 | $EFFFFFFF | 768 MB | Reserved | NO CACHE - NO BURST |
| **$F0000000** | **$FBFFFFFF** | **192 MB** | **CT60 I/O** | NO CACHE - NO BURST |
| $FC000000 | $FEFFFFFF | 48 MB | Reserved | NO CACHE - NO BURST |
| $FF000000 | $FFFFFFFF | 16 MB | FALCON 24-Bit SHADOW | NO CACHE - NO BURST |

From the 030, the FLASH chip is accessible (to program and read it) by the 030 CPU SPACE #3.

From the 060, the TOS chip is NOT accessible.
The FLASH is seen at the TOS addresses when booting.
When programming the Flash in 060 mode, the ALTERNATE SPACE #3 must be used.

## REGISTERS SUMMARY

**SDR-60**

| | | | |
|---|---|---|---|
| EE | EECL | $F0000000 | I2C port for EEprom DIMM. |
| | EEDA | $F0800000 | |
| TH | THCS | $F1000000 | THermal sensor of the 060. |
| | THCK | $F1800000 | |
| | THDA | $F1000000 | |
| SDCNF | | $F2000000 | SDram CoNFiguration. |
| IVR | | $F3000000 | Int Vector Register. |

**ABE-60**

| | | |
|---|---|---|
| FWEN | $F9000000 | Flash Write ENable. |
| SLP | $FA000000 | Sleep = Turn OFF the ATX power supply. |

## ABE60 REGISTERS DETAILS

**FWEN (Flash Write ENable)**

Only for the PowerPC write accesses – Not needed for 030 and 060 write accesses.
Write at $F9000000 → Can write
Write at $F9800000 → Can't write

**SLP (SleeP)**

Write at $FA800000 → Turn OFF the power supply.

## SDR60 REGISTERS DETAILS

### SDRAM EEPROM I2C Port :

**EECL (EEprom serial CLock)**

Write at $F0000000 → WRITE 0 to EECL line.
Write at $F0400000 → WRITE 1 to EECL line
Read at $F0000000 → READ from the EECL line on the D1 CPU data line.

**EEDA (EEprom serial DAta)**

Write at $F0800000 → WRITE 0 to EEDA line.
Write at $F0C00000 → WRITE 1 to EEDA line.
Read at $F0000000 → READ from the EEDA line on the D0 CPU data line.

### 060 THERMAL 3-wires Port :

**THCS (THermal Chip Select)**

Write at $F1000000 → WRITE 0 to CS line.
Write at $F1400000 → WRITE 1 to CS line.

**THCK (THermal ClocK)**

Write at $F1800000 → WRITE 0 to CLK line.
Write at $F1C00000 → WRITE 1 to CLK line.

**THDA (THermal DAta)**

Read at $F1000000 → Read from the DO line on the D0 CPU data line.

### INTERRUPT REGISTER :

**IVR (Interrupt Vector Register)**

Read at $F3000000

SDRAM CONTROLLER :

**SDCNF (SDram CoNFiguration)**

**Write a long at $F2xx0000 with xx = [A23..A16]**

**Chip DensitY (EEPROM Byte #3 & #4)**

A23 = cdy2
A22 = cdy1

| | | | Byte#3 | Byte#4 |
|---|---|---|---|---|
| [cdy2,cdy2] | = 0,0 | --> 8Mx8; 8x16 | $0C | $09 |
| | = 0,1 | --> 16Mx8 | $0C | $0A |
| | = 1,0 | --> 16Mx16 | $0D | $09 |
| | = 1,1 | --> 32Mx8; 32Mx16 | $0D | $0A |

**NumbeR of DIMM Banks (EEPROM Byte #5)**

A20 = nrb

| | | |
|---|---|---|
| [nrb] | = 0 | --> 1 bank |
| | = 1 | --> 2 banks |

**Module DensitY (EEPROM Byte #31 * EEPROM Byte #5)**

A19 = mdy2
A18 = mdy1

| | | |
|---|---|---|
| [mdy2,mdy1] | = 0,0 | --> 64MB |
| | = 0,1 | --> 128MB |
| | = 1,0 | --> 256MB |
| | = 1,1 | --> 512MB |

**ReFresh RaTe (EEPROM Byte #12)**

A16 = rfrt

| | | |
|---|---|---|
| [rfrt] | = 0 | --> 15.360 uS |
| | 1 | --> 7.680 uS |

# 060 BUS SLOT

A 060 bus Slot is present on the CT60 for some future daughter cards like the PCI adaptor or a PPC developers system.

A minimal and usefull connector was choosen. It has only 100 pins (2 connectors of 2x25 pins) and furnishes the following signals and power lines.

## Signals Groups

| | |
|---|---|
| ADDRESS & DATA | |
| A31-A0 | **A**ddress Bus |
| D31-D0 | **D**ata Bus |
| | |
| TRANFER CONTROL | |
| /TS | **T**ransfer **S**tart |
| R/W | **R**ead **W**rite |
| /BS0, BS1/, /BS2, /BS3 | **B**yte **S**elect |
| SIZ1, SIZ0 | **SIZ**e |
| TT1, TT0, TM2, TM1, TM0 | **T**ransfer **T**ype & **T**ransfer **M**odifier |
| /TA | **T**ransfer **A**cknowledge |
| | |
| ARBITRATION | |
| /BR | **B**us **R**equest |
| /BG | **B**us **G**rant |
| /BB | **B**us **B**usy |
| | |
| INTERRUPTS | |
| /TEA | **T**ransfer **E**rror **A**cknowledge |
| /RST | **R**e**S**e**T** |
| /I6 | **I**nterrupt **6** : Sent by the daughter card to the CT60 |
| /INT | **INT**errupt : Sent by CT60 to the daughter card for a PPC. |
| | |
| CLOCK | |
| CLK | **CL**oc**K** (CT60 clock : 64MHz or more) |
| | |
| POWER | |
| -12V, +12V | Power supplies : used by some PCI cards and the fans (+12). |
| +3.3V, +5V | Power supplies : used by components and processors.<br>Each pin can drive up to 6 Amperes. |
| GND (9) | **G**rou**ND** pins. |

## Pinout

| **Add Connector** | | **Data Connector** | |
|---|---|---|---|
| ***#A1*** | ***#A2*** | ***#D1*** | ***#D2*** |
| GND | GND | GND | GND |
| +5V | /BR | TM0 | +12V |
| /TS | /BG | TM1 | -12V |
| R/W | /BB | TM2 | +3.3V |
| SIZ0 | SIZ1 | TT0 | TT1 |
| /TA | /TEA | /BS0 | /BS1 |
| | CLK | /BS2 | /BS3 |
| /RST | GND | /I6 | /INT |
| A30 | A31 | D0 | D1 |
| .. | .. | .. | .. |
| A0 | A1 | D30 | D31 |
| GND | GND | GND | GND |
| ***#A49*** | ***#A50*** | ***#D49*** | ***#D50*** |

# HARDWARE EMULATION

The CT60 allows a **Falcon hardware emulation**.

With this Hardware Emulation Window, it is easy to implement a new chip replacing the old one of the Falcon motherboard and this at the same address(es) !
Examples :

- SUPER-VIDEL chip.
- SDRAM replacement of a part of the ST-RAM at the same addresses.
- ACIA for new PS/2 ports with a CPLD/FPGA.
- ACIA MIDI with a CPLD/FPGA.
- DSP56301 replacing 56001 at same addresses !
- FPGA emulating serial & parallel port of the Falcon (Zilog 85C30 and Yamaha).
- new SDMA for Audio.

There is a time window from the start of the 060 access to the Falcon addresses ($00xxxxxx and $FFxxxxxx) up to the start (rising edge) of the 7$^{th}$ cycle of the CLK (bus and 060 clock).

When the 060 inserts the address and TS to validate, a counter into ABE start if the address is somewhere in the Falcon address space.
Until the counter reaches the end of the 6 th cycle, a card on the 060 slot bus of the CT60 can answer to terminate the access instead of a chip of the Falcon mb (with TA/ or TEA/ or both TA/ & TEA/ for a RETRY).
This termination of the access terminates the counter and invalidates the Falcon access that was started.

At the begining of the 7$^{th}$ cycle the Falcon READ access continues and cannot be stopped. ABE drives data on the CT60 bus.
The time limit for the termination signal sampling is the end of 6 th cycle.

If you want **to use SDRAM on a daughter card** :
For 66 MHz SDRAM BURST READ you need 5-1-1-1 cycles.
The TA arrives the 5$^{th}$ cycle (first data) up to 8$^{th}$ (fourth data). This TA arrives before the end of the 6$^{th}$ cycle and the F30 access start is cancelled.
For 66MHz SDRAM BURST WRITE you need 3-1-1-1 cycles.
The TA arrives the 3$^{rd}$ cycle (first data) up to 6$^{th}$ (fourth data). This TA arrives before the end of the 6$^{th}$ cycle and the F30 access start is cancelled.

For **registers accesses on a daughter card**, you need 2 or 3 cycles.
If you want to write both to F030 mb AND your daughter card (an adress that is present on the two boards), don't send TA from the daughter card and the TA from mb will terminate the write access for you.

By example, this technic allows to write all VIDEL and SUPER VIDEL registers in the same time. The emulation is total! The only thing there is to do is to implement a bit in the daughter board to **switch ON/OFF the emulation**.

**If the switch is ON :**
- the daughter card address registers are at the same addresses than the F030 mb and :
- the TA must not be sent when writting these registers that are common to F030 and the daughter card.
- the TA must be sent before the 7$^{th}$ cycle when reading from register that is a common to F030 & daughter card.

**If the switch is OFF :**
**-** the daughter card address registers must be present at some specific addresses (not the same than the F030) and the TA is sent as usual byt the daughter card for all read & write accesses.

Example with $FFFF820E :
Switch is ON --> Write at $FFFF820E write to daughter card and F030 mb and this access is terminated by the TA from Falcon mb (ABE). The card don't send TA.
Switch is OFF --> Write at $FFFF82OE write only to Falcon mb. You need to write to a 'new' address on the card to access the same register.

# THERMAL SENSOR

The 68060 contains a Die Temperature Sensor with two external pins THERM0 & THERM1. The sensor is done with a temperature sensitive resistor which has a 780 ohms value at 25°C and increases/decreases by steps of 2.8 ohms per °C unit. By example, a 060 core at 80°C gives a resistance of 934 ohms between the two THERMx pins.
**Equations : R60 = 780 + 2.8 x (TEMP – 25) or TEMP = (R60 - 710) / 2.8**

The CT60 uses a small slow Analog/Digital converter (TI TLV0831) to obtain a 8-Bit value of the voltage between the THERMs pins.
**The equation is : U60 = (3.34 x R60) / (1000 + R60)** where R60 is the value of the core sensor resistor; 3.34 is the power supply and 1000 is the value of the resistor connected between the 3.34V and the positive THERM0/IN+ line.
**TOLERANCES** :

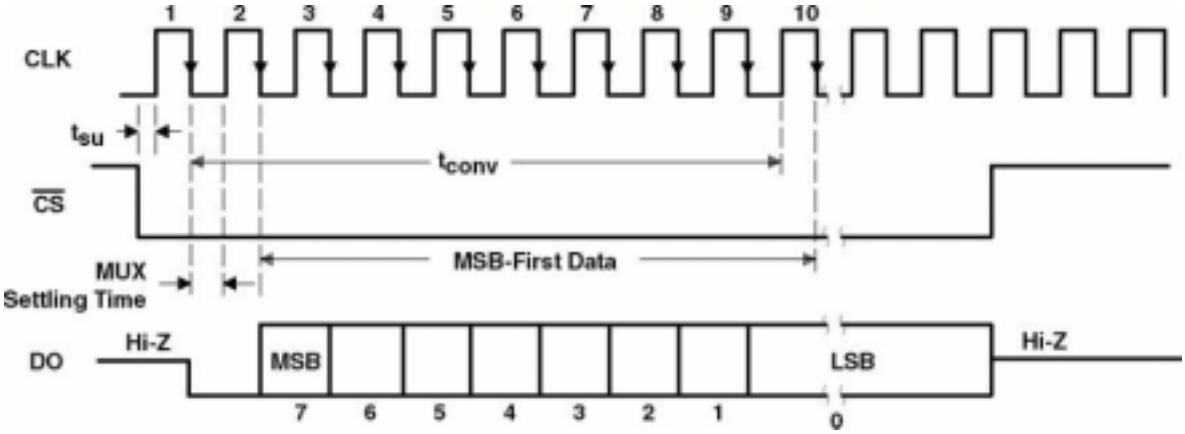
- Power supply : 3.3V +/- 4% → From 3.168V to 3.432V. Curently, it is 3.30 to 3.34.
- Resistor : 1K +/-1% → From 990 to 1010 Ohms. Curently, it is from 995 to 1005 ohms.

The AD converter uses a **REF voltage of 1.800 V**. With 0 to 0.007 V between the two pins of the AD converter, the digital result is 0. With 1.794 to 1.800 V, the result is 255. The value increases/decreases by **steps of 0.007 V**.
**The equation is : Data = INT [U60/0.007].**

| | | | |
|---|---|---|---|
| At 0°C : | Vin+ = 1.387 V | Data = 197 | R60=710 |
| At 25°C : | Vin+ = 1.464 V | Data = 208 | R60=780 |
| At 50°C : | Vin+ = 1.535 V | Data = 218 | R60=850 |
| At 100°C : | Vin+ = 1.662 V | Data = 236 | R60=990 |

**ATTENTION : The varation of the data is not linear !!**

The CPU must access the TLV831 by a basic bit-by-bit protocol. It is the software responsibility to respect the protocol & timings of the following chronogram, and assemble the bits.




| | |
|---|---|
| f : Clock frequency | 10 to 600 kHz (typical = 250) |
| tsu : Setup time, CS LOW before CLK goes HIGH | 350 ns MIN |
| tpd : Propagation delay time :output data after CLK goes HIGH | 500 ns MAX (typical = 200) |
| twh : Pulse duration, CS HIGH | 220 ns MIN |
| tconv : Conversion Time (at 250kHz) | 32 us |

Three registers are present in the SDR60 chip.
The 060 CPU must drive THCS & THCK and read THDA by these registers.
The address $F1000000, $F1800000 & $F1000000 are used respectively for THCS, THCK & THDA.

**THCS (Chip Select)**

| | | |
|---|---|---|
| LONG WRITE at $F1000000 | WRITE 0 to CS | Rising edge of CS (removed) |
| LONG WRITE at $F1400000 | WRITE 1 to CS | Falling edge of CS (active) |

**THCK (Clock)**

| | | |
|---|---|---|
| LONG WRITE at $F1800000 | WRITE 0 to CLK | Falling edge of CLK |
| LONG WRITE at $F1C00000 | WRITE 1 to CLK | Rising edge of CLK |

**THDA (Data Output)**

| | |
|---|---|
| LONG READ at $F1000000 | READ from DO – Value is available on D0 of the CPU data bus. |

For an example, see the example in the DIMM EEPROM chapter.

# DIMM EEPROM

## EEPROM DATA

The DIMM standard allows the loading of the manufacturer informations from a small 128 or 256 bytes EEPROM on the DIMM. Some of these informations are needed to configure the SDRAM controller of the CT60.

The following bytes are uses by the CT60 :

- **Bold** are used by the boot software to configure the SDRAM controller.
- Others are used only as user information in a SET UP menu.

| | | |
|---|---|---|
| Byte #2 | Memory Type | FPM; EDO; NIBBLE; **SDRAM=$04** |
| **Byte #3** | **Number of Row Addresses** | **12=$0C; 13=$0D** |
| **Byte #4** | **Number of Column Addresses** | **8=$08; 9=$09; 10=0A**; 11=$0B |
| **Byte #5** | **Number of DIMM Banks** | **1=$01; 2=$02** |
| Byte #6 & 7 | Module Data Width | **64=$4000;** 72; 80 |
| Byte #8 | Voltage Interface Level of this assembly | TTL; **LVTTL=$01;** HSTL; SSTL3; SSTL2 |
| Byte #9 | SDRAM Cycle Time (tCYC) | |
| Byte #10 | SDRAM Access from Clock (tAC) | |
| Byte #11 | SDRAM Configuration Type | **None=$00**; Parity; ECC |
| **Byte #12** | **Refresh Rate** | **15.625uS=$80; 7.81uS=$82** |
| Byte #17 | Number of Banks on SDRAM Device | 2; **4=$04** |
| Byte #27 | Minimum ROW Precharge Time (tRP) | |
| Byte #28 | Minimum ROW Active to Active Delay (tRRD) | |
| Byte #29 | Minimum RAS to CAS Delay (tRCD) | |
| **Byte #31** | **Module Bank Density** | **32=$08; 64=$10; 128=$20; 256=$40; 512=$80** |
| Byte 64-71 | Module Manufacturer's JEDEC ID Code | EX : $**A4000000 = IBM** |
| Byte 73-90 | Module Part Number | |
| Byte 93-94 | Module Manufacturing Date | |
| Byte 95-98 | Module Serial Number | |

Some features are initialized by the logic chip into the DIMM module when booting :

| | | |
|---|---|---|
| - **BURST Length** | 1, 2, **4**, 8 , Page | **4** is for **060, PPC, X86** processors |
| - **CAS Latency** | **2**, 3, 4, … | **2** is possible with **PC100** at 66 up to 80 MHz ! |

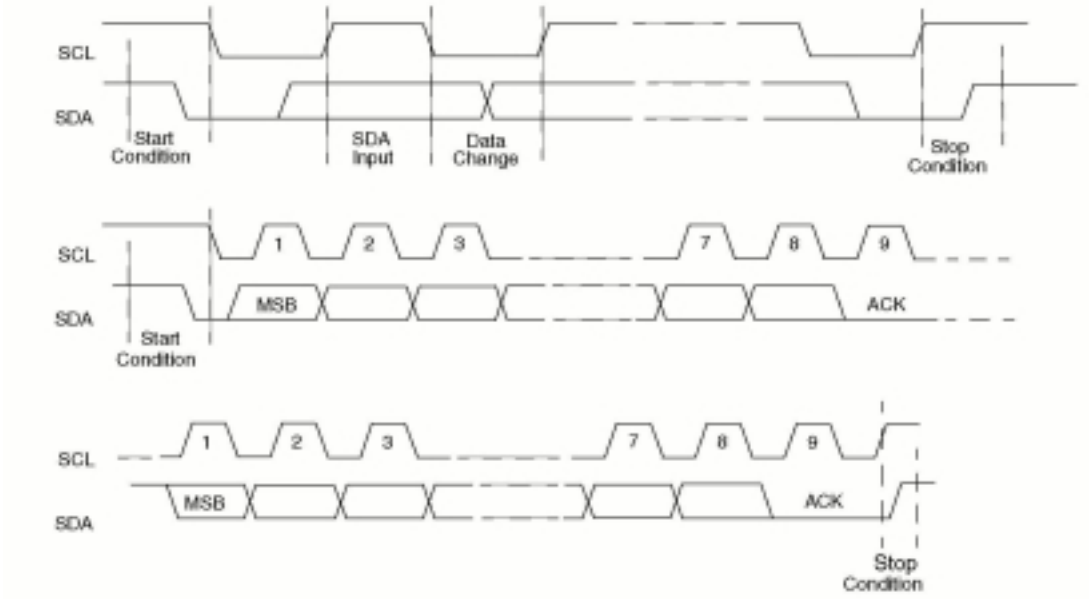
### Remarks :

**- Don't confuse SDRAM banks (2 or 4) with DIMM banks (1 or 2) !**
- Bytes 128-255 are open for Customer Use and can be written – Not used with CT60.
- DIMM Density = Module Bank Density * Number of DIMM Banks (1 or 2).

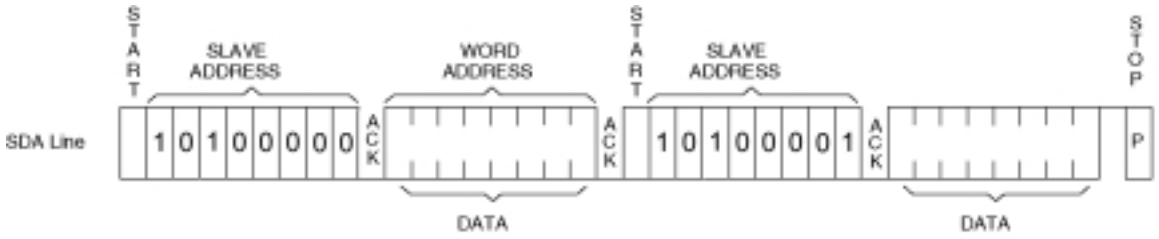
## I2C 2-wire PROTOCOL

The EEPROM device conforms to the I2C 2-wire protocol. CT60 only uses the **random read operations** with the EEPROM.

During data input, the EEPROM samples the SDA signal on the rising edge of the clock (SCL). For correct device operation, the SDA signal must be stable during the clock low to high transition and data must change only when the clock (SCL) line is low.



## RANDOM READ PROTOCOL & SOFTWARE




The slave address is 1010000. The eight bit is the R/W bit.

Random read operations allow the master to access any memory location in a random manner. Before issuing the slave address with the R/W bit set to one (Read), the master must first perform a dummy write operation. The master issues the start condition, slave address and then the word address it is to read. After the word address ACK, the master immediately re-issues the start condition and the slave address with the R/W bit set to one. This will be followed by an ACK from the slave and then by the eight bit word. The master will not ACK the transfer but will issue a stop and the slave stops transmission and goes into standby.
The device that controls the transfer is referred to as the master (SDR60 chip) and the device that receives the data (EEPROM) is referred to as the slave device. The master will always start a data transfer (SDA line) and will provide the serial clock (SCL line) for synchronization.

The 060 CPU must drive the SCL and SDA lines. These lines are connected to 2 pins of the logic chip.
The address $F00xxxxx is used for SCL and $F08xxxxx is used for SDA signal.

**SCL (Clock)**

| | | |
|---|---|---|
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |

**SDA (Data)**

| | |
|---|---|
| LONG WRITE at $F0800000 | WRITE 0 to SDA |
| LONG WRITE at $F0C00000 | WRITE 1 to SDA |
| LONG READ at $F0800000 | READ from SDA – Value is available on D0 of the CPU data bus. |

## EXAMPLE

If you want to read the Byte #3 from the EEPROM :

**START condition**

| | | |
|---|---|---|
| LONG WRITE at $F0800000 | WRITE 0 to SDA | |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**SLAVE ADDRESS (Write at 1010000)**
**Write '1'**

| | | |
|---|---|---|
| LONG WRITE at $F0C00000 | WRITE 1 to SDA | |
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**Write '0'**

| | | |
|---|---|---|
| LONG WRITE at $F0800000 | WRITE 0 to SDA | |
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**…**
**Repeat for the values 10000** (the last 0 is for 'write')

**ACK condition**

| | | |
|---|---|---|
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG READ at $F0800000 | READ from SDA | If =0, it's an ACK |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**WORD ADDRESS DATA** (# 3 in this example)
**Write '0'**

| | | |
|---|---|---|
| LONG WRITE at $F0800000 | WRITE 0 to SDA | |
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**…**
**Repeat 5 times**

**Write '1'**

| | | |
|---|---|---|
| LONG WRITE at $F0C00000 | WRITE 1 to SDA | |
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**…**
**Repeat 1 time**

**ACK condition**

| | | |
|---|---|---|
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG READ at $F0800000 | READ from SDA | If =0, it's an ACK |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**START condition immediately after ACK**

| | | |
|---|---|---|
| *LONG WRITE at $F0C00000* | *WRITE 1 to SDA* | |
| *LONG WRITE at $F0400000* | *WRITE 1 to SCL* | *Rising edge of SCL* |
| LONG WRITE at $F0800000 | WRITE 0 to SDA | |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**SLAVE ADDRESS (Read at 1010000)**
**Write '1'**

| | | |
|---|---|---|
| LONG WRITE at $F0C00000 | WRITE 1 to SDA | |
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**Write '0'**

| | | |
|---|---|---|
| LONG WRITE at $F0800000 | WRITE 0 to SDA | |
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**…**
**Repeat same procedure for the values 10001** (the last 1 is for 'read')

**ACK condition**

| | | |
|---|---|---|
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG READ at $F0800000 | READ from SDA | If =0, it's an ACK |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**READ WORD DATA**
**Bit#7**

| | | |
|---|---|---|
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG READ at $F0800000 | READ from SDA | DATA Bit#7 |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**Bit#6**

| | | |
|---|---|---|
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG READ at $F0800000 | READ from SDA | DATA Bit#6 |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**…**
**Repeat 6 times**

**Clock cycle (NO ACK)**

| | | |
|---|---|---|
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG WRITE at $F0000000 | WRITE 0 to SCL | Falling edge of SCL |

**STOP condition**

| | | |
|---|---|---|
| LONG WRITE at $F0800000 | WRITE 0 to SDA | |
| LONG WRITE at $F0400000 | WRITE 1 to SCL | Rising edge of SCL |
| LONG WRITE at $F0C00000 | WRITE 1 to SDA | |

# DIMM SDRAM for CT60

The CT60 supports the **PC100/133 standard SDRAM DIMMs**, but some obsolet/uneeded are not accepted.

**GOOD SDRAM DIMMs for CT60**
- **PC-100 & PC-133**.
- Must be **UNBUFFURED** type.
- Must be 64-bits (no parity or ECC = 72 / 80 bits).

| **Module Config.** | **CHIPS / Side** | **SIDES** (1=single) (2=double) | **CHIPS Archit.** | **RAS Addr.** | **CAS Addr.** | **PAGE Lentgh** (4 banks) | **Refresh Rate** (uS) |
|---|---|---|---|---|---|---|---|
| **64 MB** | 8 | 1 | 8Mx8b | 12 | 9 | 8 KB | 15.625 |
| **64 MB** | 4 | 1 | 8Mx16b | 12 | 9 | 8 KB | 15.625 |
| **128 MB** | 8 | 2 | 8Mx8b | 12 | 9 | 8 KB | 15.625 |
| **128 MB** | 4 | 2 | 8Mx16b | 12 | 9 | 8 KB | 15.625 |
| **128 MB** | 8 | 1 | 16Mx8b | 12 | 10 | **16 KB** | 15.625 |
| **128 MB** | 4 | 1 | 16Mx16b | 13 | 9 | 8 KB | 7.8125 |
| **256 MB** | 8 | 2 | 16Mx8b | 12 | 10 | **16 KB** | 15.625 |
| **256 MB** | 4 | 2 | 16Mx16b | 13 | 9 | 8 KB | 7.8125 |
| **256 MB** | 8 | 1 | 32Mx8b | 13 | 10 | **16 KB** | 7.8125 |
| **512 MB** | 8 | 2 | 32Mx8b | 13 | 10 | **16 KB** | 7.8125 |
| **512 MB** | 4 | 2 | 32Mx16b | 13 | 10 | **16 KB** | 7.8125 |

**NOT SUPPORTED SDRAM DIMMs**
**- All DIMM with chips density < 64Mbits :**
  **- 8MB, 16MB & 32MB** DIMMs.
  **- 64MB** DIMMs **with 16 CHIPS and / or with chips on the 2 sides.**
  - All DIMMs with **2 logical banks chips** = obsolet (CT60 needs 4 logical banks chips).
**- REGISTERED / BUFFURED** DIMMs (generally for Work Stations & Servers, not PC).
- 512 MB DIMM with one physical bank (only 1 side populated).

**REMARKS :**
**- Don't confuse logical banks (2 or 4) with physical banks (1=Single Size or 2=Double Size) !**

**PERFORMANCES :**

**The better system performances is obtained with 16 KB page lentgh DIMMs.**

# 060 BURST with SDRAM

The CT60 bus clock = the 060 clock (060 in 'Full Bus mode').
The 060 uses **LINE BURST** to & from the system memory. SDRAM is well adapted for a such processor !
A BURST LINE is a length of **4 LONG-WORDs (16 Bytes)** that are transferred with only :

**PAGE HIT** (access to a logical SDRAM page already open) :
3,1,1,1 = 6 cycles for Burst Writes. Rate is 16 Bytes / 6 cycles = 178 MBytes/s (Each access in the same page).
5,1,1,1 = 8 cycles for Burst Reads. Rate is 16 Bytes / 8 cycles = 132 MBytes/s (Each access in the same page).

**PAGE MISS (access to a new logical SDRAM page (must be precharged and open) :**
7,1,1,1 = 10 cycles for Burst Writes. Rate is 16 Bytes / 10cycles = 107 MBytes/s (Each access in a new page).
9,1,1,1 = 12 cycles for Burst Reads. Rate is 16 Bytes / 12cycles = 89 MBytes/s (Each access in a new page).

The CT60 uses the 060 at the top of the possible performances with the mighty **COPYBACK** mode ! Instead of the WRITETROUGH mode like other TOS machines !

**Copyback mode is active for all SDRAM memory area**.
Copyback mode allows the 060 to write into the cache without writting into the SDRAM, what is so more performant ! The cache lines are pushed into SDRAM only when needed (060 needs place by example). With **two 8Kbytes caches**, it gives to coders the possibility to do some incredibly speedy routs residing at 100% into the caches !

The 060 uses BURST transfers with SDRAM in 99% of the cases. Here are the cases when the 060 don't burst, this means, transfers Bytes, Words & Long-Words :

**Byte, Word, and Long-Word READ Transfer Cycles from SDRAM**

Accesses that are implicitly NONCACHABLE :
- Locked Read-Modify-Write accesses.
- Table Searches.

Accesses that are not allocate in the data cache on a read miss :
- Exception Vector Fetches.
- Exception stack Deallocation for an RTE Instruction.

**Byte, Word, and Long-Word WRITE Transfer Cycles to SDRAM**

Accesses that are implicitly NONCACHABLE :
- Locked Read-Modify-Write accesses.
- Table Searches.

Accesses that are not allocate in the data cache on a write miss :
- Exception stacking.

Cache Line pushes for lines containing a single dirty Long-word.

Write to WRITETHROUGH pages (ST-RAM !).

**Remark :**
For those of you who are a bit familiar with 64-bit processors like PPC or X86, don't forget that the syntax for data size is not the same :

**With 32-Bit processors :**
- A **WORD** designates a **16-Bit entity**.
- **LONG-WORD** designates a **32-Bit entity**.

**With 64-Bit processors :**
- A **HALF-WORD** designates a **16-Bit entity**.
- A **WORD** designates a **32-Bit entity**.
- A DOUBLE-WORD designates a 64-Bit entity.

# INTERRUPTS

CT60 adds some new interrupts for the 060 Bus Slot, re-routing of the Falcon INT Set to the second CPU (CPU#2 on the Bus Slot).

**The /I6 , and /INT were added on CT60**.

**/I6** is the interrupt from the 060 BUS SLOT and is merged with the others from the Falcon. See table below for the priority position.

**/INT** is sent by SDR-60 to the CPU#2 to interrupt it. INT is synthetized from the /IPL2, /IPL1 & /IPL0 signals and the I6. It is necessary when the CPU#2 has to respond the interrupts instead of the primary 060…

## 060 INTERRUPTS PRIORITY TABLE

| NAME | LEVEL | ACTIVE | TYPE | SOURCE | IVR | PRIORITY |
|---|---|---|---|---|---|---|
| **I6** | **6** | **Low** | **Software** | **CT60 Bus Slot** | **1, 1, 1** | **Highest** |
| INT6 | 6 | Low | Software | F030 Bus Slot | 1, 1, 0 | |
| MFPINT | 6 | Low | Software | F030 MFP | 1, 1, 0 | |
| DSPREQ | 6 | Low | Software | F030 DSP | 1, 1, 0 | |
| INT5 | 5 | Low | Software | F030 SCC | 1, 0, 1 | |
| VBL | 4 | Low | Auto | F030 VIDEL VSync | 1, 0, 0 | |
| INT3 | 3 | High | Software | F030 Bus Slot | NOT USED | |
| HBL | 2 | Low | Auto | F030 VIDEL HSync | 0, 1, 0 | |
| INT1 | 1 | High | Software | F030 Bus Slot | NOT USED | **Lowest** |

**INT1 & INT3 are NO MORE SUPPORTED with CT60 !**
INT6 is also named MFPINT on atari documents because it is daisy chained with the MFP.

To allow the PPC to read the level of the falcon re-routed interrupts, the CT60 furnishes a register called **IVR (Interrupt Vector Register).** The IVR column gives you the binary values encoded by the SDR60 chip…
Note that the INT6, MFPINT and DSPREQ Interrupts are chained on the ‘level 6’ line…(it’s a stock Falcon feature !).

It is planned that the PPC board will contain a mechanism register to generate a ‘68K like’ INT ACK cycle and receive the software Vector from the Falcon data bus. On CT60, the IVR contains only the level of the pending INT…

## CPU#2 INTERRUPT

| NAME | LEVEL | ACTIVE | TYPE | SOURCE |
|---|---|---|---|---|
| **INT** | None | **Low** | **Auto** | **F030 IPLx and CT60 I6** |

This interrupt is compatible with the PowerPC INT…

# CHIPSET PIN-OUT

## SDR-60

| | |
|---|---|
| 1 VCC | 73 VCC |
| 2 rstf | 74 cs1 |
| 3 PGND | 75 cs0 |
| 4 ta | 76 cas |
| 5 PGND | 77 we |
| 6 PGND | 78 a31 |
| 7 PGND | 79 a30 |
| 8 VCC | 80 a29 |
| 9 PGND | 81 a28 |
| 10 a10 | 82 a27 |
| 11 a11 | 83 a26 |
| 12 a12 | 84 VCC |
| 13 a13 | 85 a25 |
| 14 a14 | 86 a24 |
| 15 a15 | 87 a23 |
| 16 a2 | 88 a22 |
| 17 a3 | 89 GND |
| 18 GND | 90 GND |
| 19 a4 | 91 a21 |
| 20 a5 | 92 a20 |
| 21 a6 | 93 a19 |
| 22 a7 | 94 a18 |
| 23 a8 | 95 a17 |
| 24 a9 | 96 a16 |
| 25 dm3 | 97 tbi |
| 26 dm1 | 98 rst60 |
| 27 dm2 | 99 GND |
| 28 dm0 | 100 TEST SDR |
| 29 GND | 101 TEST SDR |
| 30 clk500 | 102 tci |
| 31 cs3 | 103 PGND |
| 32 clk | 104 PGND |
| 33 PGND | 105 PGND |
| 34 PGND | 106 int |
| 35 cs2 | 107 PGND |
| 36 GND | 108 GND |
| 37 VCC | 109 VCC |
| 38 PGND | 110 PGND |
| 39 ideled | 111 ipl2f |
| 40 d2 | 112 ipl1 |
| 41 d1 | 113 i6 |
| 42 VCC | 114 GND |
| 43 d0 | 115 ipl2 |
| 44 PGND | 116 bs1 |
| 45 PGND | 117 ts |
| 46 ma12 | 118 tt1 |
| 47 GND | 119 PGND |
| 48 ma11 | 120 PGND |
| 49 ba1 | 121 ipl0 |
| 50 ba0 | 122 TDO |
| 51 ma10 | 123 GND |
| 52 ma9 | 124 siz1 |
| 53 ma8 | 125 rsto |
| 54 ma7 | 126 eeda |
| 55 VCC | 127 VCC |
| 56 ma6 | 128 thcs |
| 57 ma5 | 129 rw |
| 58 ma4 | 130 siz0 |
| 59 ma3 | 131 Reserved |
| 60 ma2 | 132 PGND |
| 61 ma1 | 133 PGND |
| 62 GND | 134 eecl |
| 63 TDI | 135 bs0 |
| 64 ma0 | 136 bs2 |
| 65 TMS | 137 bs3 |
| 66 PGND | 138 thck |
| 67 TCK | 139 ipl0f |
| 68 PGND | 140 ipl1f |
| 69 PGND | 141 VCC |
| 70 PGND | 142 thdi |
| 71 ras | 143 rst |
| 72 GND | 144 GND |

## ABE-60

| | |
|---|---|
| 1 VCC | 73 VCC |
| 2 avec | 74 d28 |
| 3 a13 | 75 d27 |
| 4 ct60 | 76 d26 |
| 5 a14 | 77 d25 |
| 6 a15 | 78 d24 |
| 7 ta | 79 d23 |
| 8 VCC | 80 d22 |
| 9 bs0 | 81 d21 |
| 10 flhoe | 82 d20 |
| 11 flhwe | 83 d19 |
| 12 dtkcmb | 84 VCC |
| 13 halt | 85 d18 |
| 14 a2 | 86 d17 |
| 15 a3 | 87 d16 |
| 16 tt0 | 88 d15 |
| 17 tm0 | 89 GND |
| 18 GND | 90 GND |
| 19 bg1 | 91 d14 |
| 20 tm1 | 92 d13 |
| 21 bg2 | 93 d12 |
| 22 ts | 94 d11 |
| 23 bg0 | 95 d10 |
| 24 tt1 | 96 d9 |
| 25 exp/ - TEST ABE | 97 d8 |
| 26 exp2/ - TEST ABE | 98 d7 |
| 27 tm2 | 99 GND |
| 28 as | 100 d6 |
| 29 GND | 101 d5 |
| 30 clk500 | 102 d4 |
| 31 bs1 | 103 d3 |
| 32 clk | 104 d2 |
| 33 bs2 | 105 d1 |
| 34 i6 | 106 br2 |
| 35 dtk | 107 d0 |
| 36 GND | 108 GND |
| 37 VCC | 109 VCC |
| 38 clkf | 110 fd15 |
| 39 a2f | 111 fd14 |
| 40 fc2 | 112 fd13 |
| 41 slp | 113 fd12 |
| 42 VCC | 114 GND |
| 43 bs3 | 115 fd11 |
| 44 a3f | 116 fd10 |
| 45 bb | 117 fd9 |
| 46 fc1 | 118 fd8 |
| 47 GND | 119 fd7 |
| 48 a16 | 120 fd6 |
| 49 a17 | 121 fd5 |
| 50 a18 | 122 TDO |
| 51 a19 | 123 GND |
| 52 a20 | 124 fd4 |
| 53 a21 | 125 fd3 |
| 54 a22 | 126 fd2 |
| 55 VCC | 127 VCC |
| 56 a23 | 128 fd1 |
| 57 a24 | 129 fd0 |
| 58 a25 | 130 bg30 |
| 59 a26 | 131 uds |
| 60 a27 | 132 lds |
| 61 a28 | 133 a1f |
| 62 GND | 134 berr |
| 63 TDI | 135 br0 |
| 64 a29 | 136 rw |
| 65 TMS | 137 fc0 |
| 66 a30 | 138 br1 |
| 67 TCK | 139 abdir |
| 68 a31 | 140 bgk |
| 69 d31 | 141 VCC |
| 70 d30 | 142 tea |
| 71 d29 | 143 rst |
| 72 GND | 144 GND |