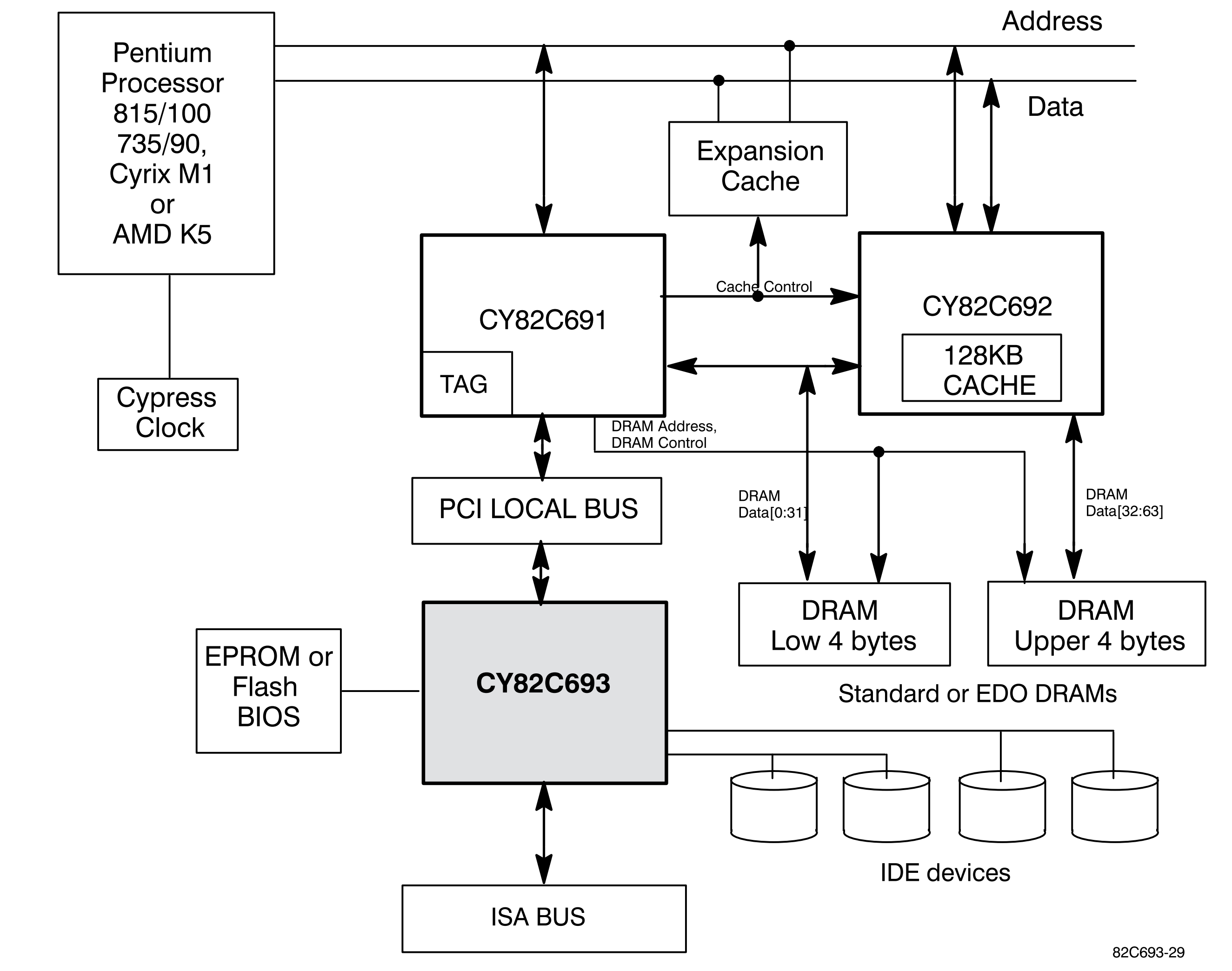
ADVANCED INFORMATION **CY82C693**

Pentium™ hyperCache™ Chipset Peripheral Controller

Features

- PCI to ISA bridge
- PCI Bus Rev. 2.1 compliant
- Supports up to 5 additional PCI masters including the CY82C691
- Integrated DMA controllers with Type A, B, and F support.
- Integrated Interrupt controllers
- Integrated timer/counters
- Integrated Real-Time-Clock with 242 bytes of additional battery-backed SRAM
- Write-only Register Shadowing
- Integrated Dual-Channel enhanced IDE controller with
  - — CD ROM support
  - — PIO modes 0 through 4 operation
  - — Single-word and Multi-word DMA modes 0 through 2
  - — PCI bus mastering
- Integrated Keyboard Controller
- APM compliant power management support through SMM or under hardware control.
- Flash PROM support with Write-protection
- Power-on reset circuitry
- QuietBus™ support for the PCI and ISA bus interfaces for better noise immunity
- General-purpose I/O pins and registers
- Zero TTL system operation
- Provides PCI-ISA/ISA-PCI/ IDE-PCI/PCI-IDE post writing
- Provides ISA-PCI pre-reading
- Packaged in a 208-pin PQFP

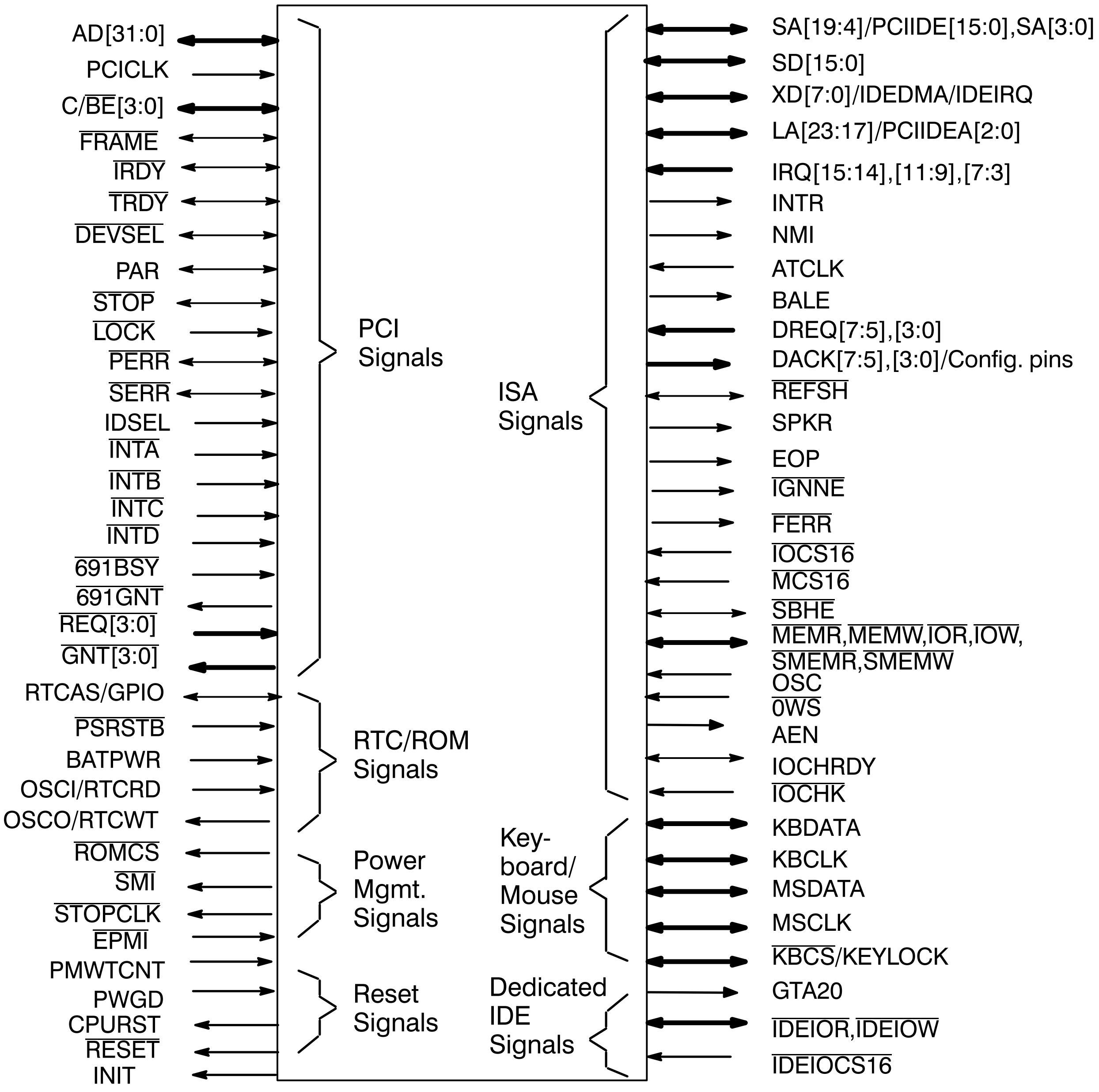
System Block Diagram



Pentium is a trademark of Intel Corporation.
hyperCache and QuietBus are trademarks of Cypress Semiconductor Corporation.

June 1995

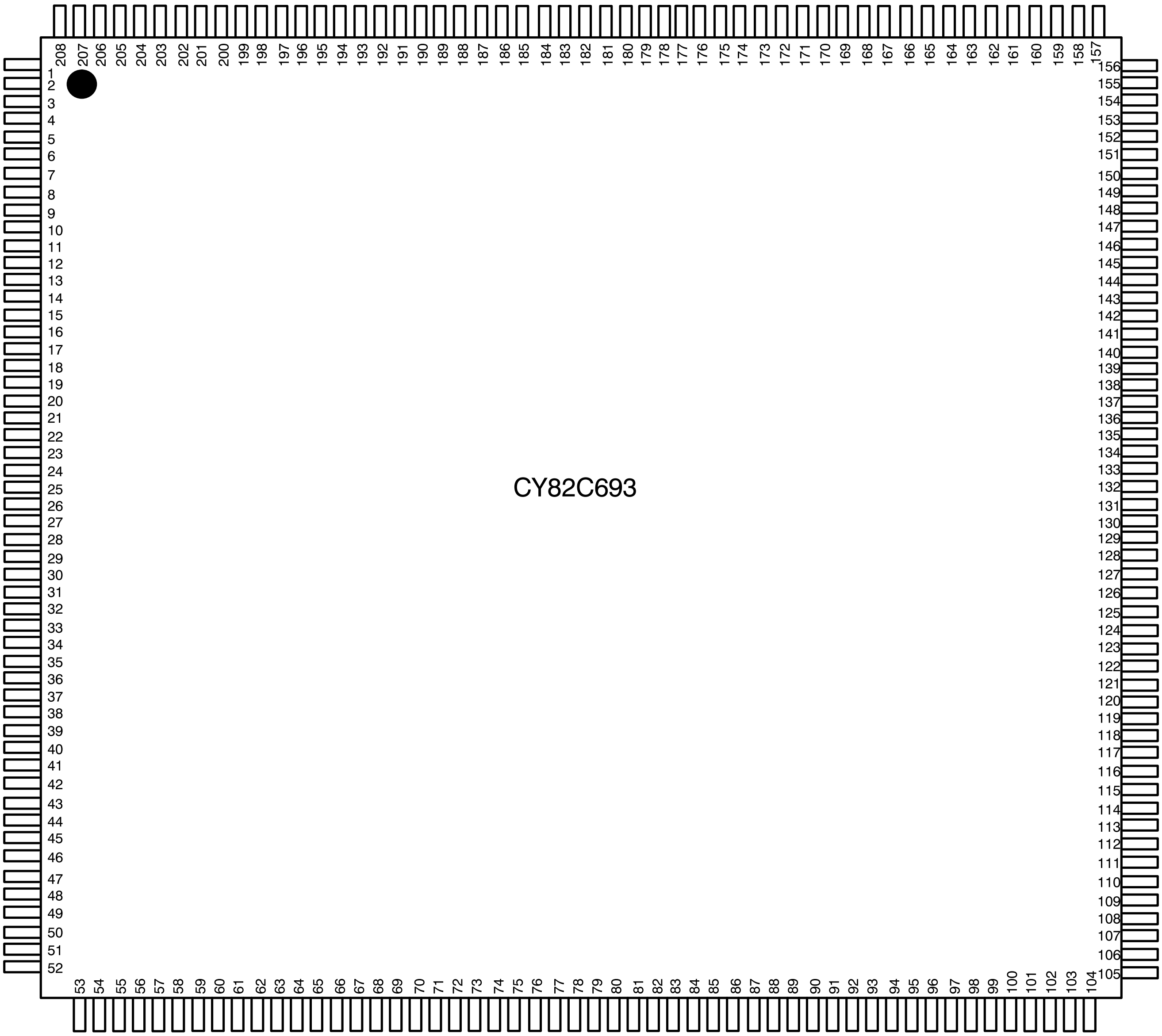
CY82C693 Signals



82C693-30

Pin Configuration

208-pin PQFP
Top View



82C693-31

CY82C693 Pin Reference (In Numerical Order by Pin Number)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	GND	43	AD9	85	SA7	127	OSCI	169	+3.3V
2	$\overline{\text{FRAME}}$	44	AD8	86	SA6	128	PWGD	170	$\overline{\text{EPMI}}$
3	IDSEL	45	AD7	87	SA5	129	DREQ0	171	GND
4	$\overline{\text{TRDY}}$	46	AD6	88	SA4	130	DREQ5	172	PMWTCNT
5	$\overline{\text{IRDY}}$	47	AD5	89	SA3	131	DREQ6	173	CPURST
6	$\overline{\text{DEVSEL}}$	48	AD4	90	SA2	132	DREQ7	174	INIT
7	$\overline{\text{SERR}}$	49	AD3	91	SA1	133	OSC	175	NMI
8	$\overline{\text{PERR}}$	50	AD2	92	SA0	134	SD0	176	INTR
9	PAR	51	AD1	93	GND	135	SD1	177	$\overline{\text{IGNNE}}$
10	$\overline{\text{LOCK}}$	52	AD0	94	$\overline{\text{ROMCS}}$	136	+5V	178	$\overline{\text{FERR}}$
11	C/$\overline{\text{BE}}$3	53	GND	95	IRQ7	137	GND	179	$\overline{\text{SMI}}$
12	C/$\overline{\text{BE}}$2	54	$\overline{\text{IOCHK}}$	96	IRQ6	138	SD2	180	$\overline{\text{STOPCLK}}$
13	C/$\overline{\text{BE}}$1	55	IOCHRDY	97	IRQ5	139	SD3	181	GTA20
14	C/$\overline{\text{BE}}$0	56	$\overline{\text{0WS}}$	98	IRQ4	140	SD4	182	RTCAS
15	GND	57	AEN	99	IRQ3	141	SD5	183	MSDATA
16	AD31	58	$\overline{\text{SMEMW}}$	100	IRQ1	142	SD6	184	$\overline{\text{KBCS}}$
17	AD30	59	$\overline{\text{SMEMR}}$	101	$\overline{\text{REFSH}}$	143	SD7	185	KBDATA
18	AD29	60	$\overline{\text{IOW}}$	102	IRQ9	144	$\overline{\text{MEMR}}$	186	$\overline{\text{IDEIOCS16}}$
19	AD28	61	$\overline{\text{IOR}}$	103	BALE	145	$\overline{\text{MEMW}}$	187	$\overline{\text{IDEIOW}}$
20	+5V	62	LA23	104	+5V	146	SD15	188	$\overline{\text{IDEIOR}}$
21	GND	63	LA22	105	GND	147	SD14	189	$\overline{\text{691GNT}}$
22	AD27	64	LA21	106	IRQ10	148	SD13	190	$\overline{\text{691BSY}}$
23	AD26	65	+5V	107	ATCLK	149	SD12	191	$\overline{\text{GNT0}}$
24	AD25	66	LA20	108	IRQ11	150	SD11	192	$\overline{\text{GNT1}}$
25	AD24	67	GND	109	IRQ12	151	GND	193	$\overline{\text{GNT2}}$
26	AD23	68	LA19	110	IRQ14	152	SD10	194	$\overline{\text{GNT3}}$
27	AD22	69	LA18	111	IRQ15	153	SD9	195	+5V
28	AD21	70	LA17	112	$\overline{\text{DACK0}}$	154	SD8	196	$\overline{\text{REQ0}}$
29	AD20	71	GND	113	$\overline{\text{DACK1}}$	155	XD0	197	GND
30	AD19	72	SA19	114	$\overline{\text{DACK2}}$	156	+5V	198	$\overline{\text{REQ1}}$
31	AD18	73	SA18	115	$\overline{\text{DACK3}}$	157	GND	199	$\overline{\text{REQ2}}$
32	+5V	74	SA17	116	$\overline{\text{DACK5}}$	158	XD1	200	$\overline{\text{REQ3}}$
33	GND	75	SA16	117	$\overline{\text{DACK6}}$	159	XD2	201	$\overline{\text{PCIRST}}$
34	AD17	76	SA15	118	$\overline{\text{DACK7}}$	160	XD3	202	$\overline{\text{INTD}}$
35	AD16	77	SA14	119	EOP	161	XD4	203	$\overline{\text{INTC}}$
36	AD15	78	SA13	120	IRQ8	162	XD5	204	$\overline{\text{INTB}}$
37	AD14	79	SA12	121	DREQ3	163	XD6	205	$\overline{\text{INTA}}$
38	AD13	80	SA11	122	DREQ2	164	XD7	206	$\overline{\text{STOP}}$
39	AD12	81	GND	123	DREQ1	165	$\overline{\text{SBHE}}$	207	PCICLK
40	AD11	82	SA10	124	VBATT	166	$\overline{\text{MCS16}}$	208	+5V
41	AD10	83	SA9	125	GND	167	$\overline{\text{IOCS16}}$		
42	GND	84	SA8	126	OSCO	168	SPKR		

CY82C693 Pin Reference (In Alphabetical Order by Signal Name)

Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
AD0	52	C/$\overline{\text{BE}}$2	12	IOCHRDY	55	PCICLK	207	SD4	140
AD1	51	C/$\overline{\text{BE}}$3	11	$\overline{\text{IOCS16}}$	167	$\overline{\text{PCIRST}}$	201	SD5	141
AD2	50	$\overline{\text{DACK2}}$	114	$\overline{\text{IOR}}$	61	$\overline{\text{PERR}}$	8	SD6	142
AD3	49	$\overline{\text{DACK3}}$	115	$\overline{\text{IOW}}$	60	PMWTCNT	172	SD7	143
AD4	48	$\overline{\text{DACK5}}$	116	$\overline{\text{IRDY}}$	5	PWGD	128	SD8	154
AD5	47	$\overline{\text{DACK6}}$	117	IRQ1	100	$\overline{\text{REFSH}}$	101	SD9	153
AD6	46	$\overline{\text{DACK7}}$	118	IRQ3	99	$\overline{\text{REQ0}}$	196	SD10	152
AD7	45	$\overline{\text{DEVSEL}}$	6	IRQ4	98	$\overline{\text{REQ1}}$	198	SD11	150
AD8	44	DREQ0	129	IRQ5	97	$\overline{\text{REQ2}}$	199	SD12	149
AD9	43	DREQ1	123	IRQ6	96	$\overline{\text{REQ3}}$	200	SD13	148
AD10	41	DREQ2	122	IRQ7	95	$\overline{\text{ROMCS}}$	94	SD14	147
AD11	40	DREQ3	121	IRQ8	120	RTCAS	182	SD15	146
AD12	39	DREQ5	130	IRQ9	102	SA0	92	$\overline{\text{SERR}}$	7
AD13	38	DREQ6	131	IRQ10	106	SA1	91	$\overline{\text{SMEMR}}$	59
AD14	37	DREQ7	132	IRQ11	108	SA2	90	$\overline{\text{SMEMW}}$	58
AD15	36	EOP	119	IRQ12	109	SA3	89	$\overline{\text{SMI}}$	179
AD16	35	$\overline{\text{EPMI}}$	170	IRQ14	110	SA4	88	SPKR	168
AD17	34	$\overline{\text{FERR}}$	178	IRQ15	111	SA5	87	$\overline{\text{STOP}}$	206
AD18	31	$\overline{\text{FRAME}}$	2	$\overline{\text{KBCS}}$	184	SA6	86	$\overline{\text{STOPCLK}}$	180
AD19	30	GND	1,15,21,33,42,53,67, 71,81,93,105,125, 137,151,157,171,197	KBDATA	185	SA7	85	$\overline{\text{TRDY}}$	4
AD20	29	$\overline{\text{GNT0}}$	191	LA17	70	SA8	84	VBATT	124
AD21	28	$\overline{\text{GNT1}}$	192	LA18	69	SA9	83	XD0	155
AD22	27	$\overline{\text{GNT2}}$	193	LA19	68	SA10	82	XD1	158
AD23	26	$\overline{\text{GNT3}}$	194	LA20	66	SA11	80	XD2	159
AD24	25	GTA20	181	LA21	64	SA12	79	XD3	160
AD25	24	$\overline{\text{IDEIOCS16}}$	186	LA22	63	SA13	78	XD4	161
AD26	23	$\overline{\text{IDEIOR}}$	188	LA23	62	SA14	77	XD5	162
AD27	22	$\overline{\text{IDEIOW}}$	187	$\overline{\text{LOCK}}$	10	SA15	76	XD6	163
AD28	19	IDSEL	3	$\overline{\text{MCS16}}$	166	SA16	75	XD7	164
AD29	18	$\overline{\text{IGNNE}}$	177	$\overline{\text{MEMR}}$	144	SA17	74	$\overline{\text{0WS}}$	56
AD30	17	INIT	174	$\overline{\text{MEMW}}$	145	SA18	73	$\overline{\text{691BSY}}$	190
AD31	16	$\overline{\text{INTA}}$	205	MSDATA	183	SA19	72	$\overline{\text{691GNT}}$	189
AEN	57	$\overline{\text{INTB}}$	204	NMI	175	$\overline{\text{SBHE}}$	165	+3.3V	169
BALE	103	$\overline{\text{INTC}}$	203	OSC	133	SD0	134	+5V	20,32,65,104, 136,156,195,208
ATCLK	107	$\overline{\text{INTD}}$	202	OSCI	127	SD1	135		
C/$\overline{\text{BE}}$0	14	INTR	176	OSCO	126	SD2	138		
C/$\overline{\text{BE}}$1	13	$\overline{\text{IOCHK}}$	54	PAR	9	SD3	139		

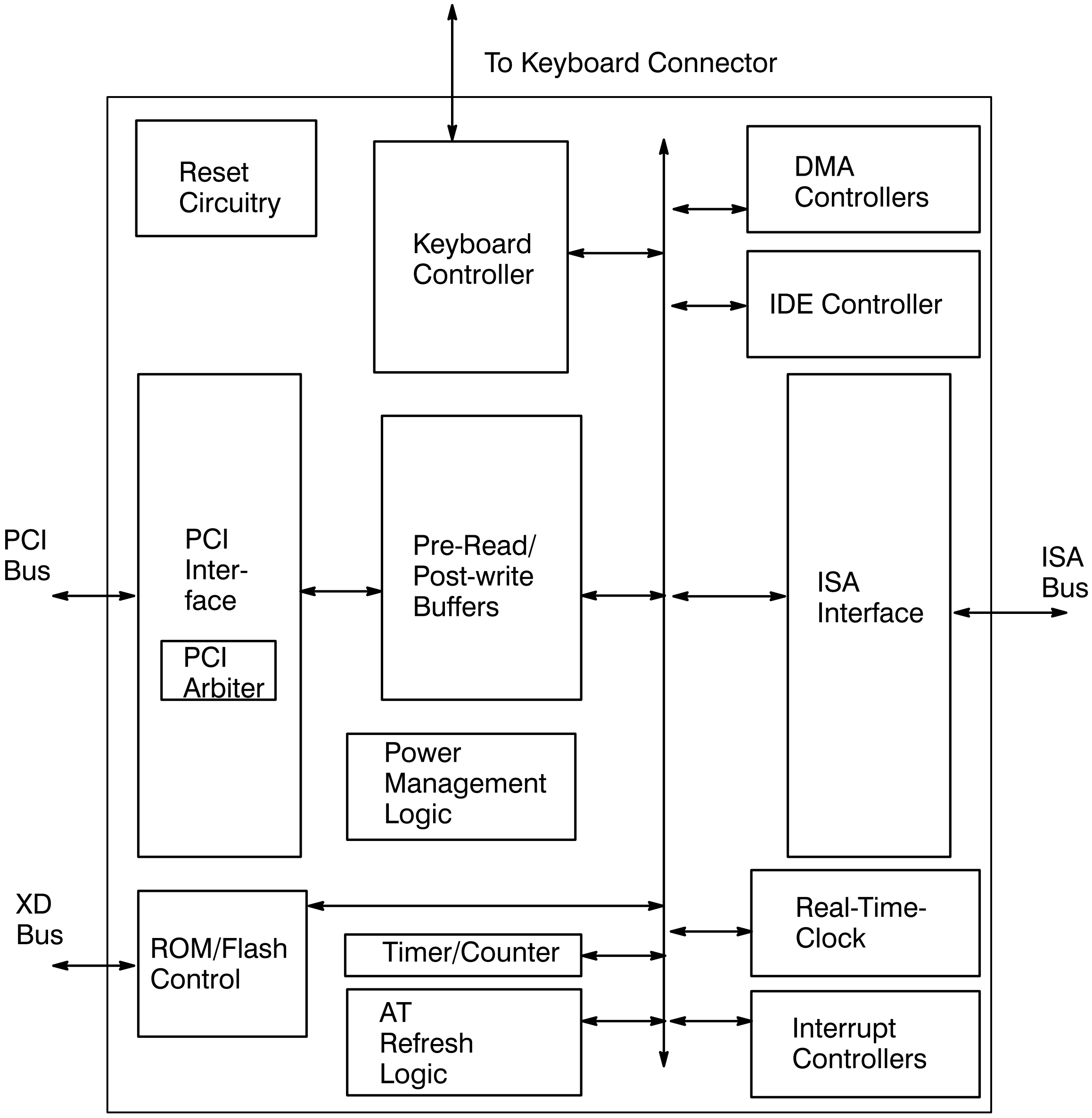


Figure 1. CY82C693 Functional Block Diagram

Introduction

System Overview

The CY82C691/692/693 hyperCache™ Chipset provides all the functions necessary to implement a 3.3V Pentium-class processor based system with the PCI (Peripheral Component Interconnect) bus and the ISA (Industry Standard Architecture) bus. The chipset consists of the CY82C691 System Controller, the CY82C692 Data Path/Cache chip, and the CY82C693 Peripheral Controller. System designers can exploit the advantages of the PCI bus while maintaining access to the large base of ISA cards in the marketplace.

The Cypress hyperCache Chipset offers system designers several key advantages. With only 3 chips, CY82C691/692/693, a complete system with a 128-KB, two-way set associative, synchronous, pipelined L2 cache can be implemented. The cache size may be increased up to 1MB with additional CY82C694 16Kx64 SRAMs or other synchronous or pipelined burst SRAMs. Six banks of page-mode or EDO DRAM further increase the system designer's options. The chipset also contains concurrent bus support, PCI enhanced IDE with CD-ROM support, integrated RTC, integrated peripheral control (Interrupts/DMA) and integrated keyboard controller. This chipset is flexible enough to provide the system designer with many cost/performance/function options to provide an optimum solution for a given design.

CY82C693 Introduction

The CY82C693 Peripheral Controller provides a highly integrated peripheral solution for PCI-based motherboards. The CY82C693 contains a PCI to ISA bridge, a PCI IDE controller, DMA controllers, Interrupt controllers, a Real-Time-Clock, and a Keyboard controller.

Figure 1 shows a block diagram of the CY82C693.

Functional Overview

The CY82C693 Peripheral Controller contains the following functional blocks:

— PCI Interface
— ISA Interface
— Reset Logic
— Keyboard Controller
— Power Management Logic
— AT Refresh Logic
— Pre-Read/Post-Write Buffers
— BIOS ROM Control.
— Timer/Counter Logic
— DMA Controllers
— Dual-Channel Enhanced IDE Controller
— Real-Time-Clock with 32 kHz Oscillator
— Interrupt Controllers

PCI Bus Interface

The CY82C693 provides a bridge for transactions between the PCI bus, the ISA bus, and IDE peripherals. PCI bus speeds of 25, 30 or 33 MHz are supported. The PCI interface is master/slave (it can initiate or be a target for transactions). The PCI bus interface in the CY82C693 is Revision 2.1 compliant. This standard allows for a multitude of high-speed peripheral cards to be added to the system. PCI is the predominant local bus for Pentium systems.

Master cycles are initiated by driving $\overline{\text{FRAME}}$ LOW with a valid address on AD[31:0], valid even address parity on PAR, and a valid command on C/$\overline{\text{BE}}$[3:0]. Data phases occur when $\overline{\text{IRDY}}$ (initiator ready) and $\overline{\text{TRDY}}$ (target ready) are both active, valid data is placed on AD[31:0], the PAR signal is driven to reflect even parity, and the correct byte enable combination is present on C/$\overline{\text{BE}}$[3:0]. Wait states can be inserted into a transaction if the initiator deasserts $\overline{\text{IRDY}}$ or the target deasserts $\overline{\text{TRDY}}$. A transaction is terminated by the deassertion of $\overline{\text{FRAME}}$ prior to the final data phase. As a PCI master, the CY82C693 will only perform memory read and write transactions. A write cycle consists of a maximum of 4 bytes of data in a single data cycle. A read consists of a maximum of 8 bytes in a two cycle burst.

The CY82C693 uses a subtractive decode strategy to determine if a PCI target transaction is destined for the ISA bus. Potential PCI targets decode any valid address during the first cycle of the transaction ($\overline{\text{FRAME}}$ asserted). If a PCI peripheral device (including the integrated IDE controller in the CY82C693) detects a transaction to its address space, it will assert $\overline{\text{DEVSEL}}$. If the CY82C693 does not detect the assertion of $\overline{\text{DEVSEL}}$ by another target (or its own IDE controller) within a programmable number of PCI clock cycles (4 or 5), it will claim the transaction by asserting $\overline{\text{DEVSEL}}$. All transactions that are not claimed by a PCI peripheral are, by default, sent to the ISA bus. After asserting $\overline{\text{DEVSEL}}$, the CY82C693 will initiate an ISA cycle and assert $\overline{\text{TRDY}}$ to the PCI bus when valid data is available.

As a PCI slave, the CY82C693 will target-terminate the cycle after the first data transfer by asserting $\overline{\text{STOP}}$ with $\overline{\text{TRDY}}$ if $\overline{\text{FRAME}}$ is not deasserted before the data phase. The CY82C693 does not accept bursts as a target. By not allowing target bursts, PCI bus bandwidth, which would otherwise be quickly consumed by ISA or IDE targets, is conserved.

The CY82C693 also supports atomic slave transactions by monitoring the $\overline{\text{LOCK}}$ signal. If the CY82C693 detects $\overline{\text{LOCK}}$ asserted in the cycle following the address phase, it will not respond to any master until $\overline{\text{LOCK}}$ is deasserted during a CY82C693 target address phase. If a different master attempts to access the CY82C693 while it is locked, the CY82C693 will keep $\overline{\text{TRDY}}$ deasserted and assert $\overline{\text{STOP}}$ along with $\overline{\text{DEVSEL}}$ to force the new initiator to end the transaction.

The CY82C693 supports PCI error reporting through the $\overline{\text{PERR}}$ (parity error) signal and the $\overline{\text{SERR}}$ (system error) signal. Both internally detected and externally generated errors are reported. $\overline{\text{PERR}}$ is asserted if a data parity error is detected, provided that the cycle is not a PCI Special Cycle. $\overline{\text{SERR}}$ is asserted for any other system error, including address parity errors or Special Cycle parity errors. If the CY82C693 detects the assertion of $\overline{\text{PERR}}$ or $\overline{\text{SERR}}$, it will assert NMI (non-maskable interrupt) to the CPU. The CY82C693 will store the source of the NMI ($\overline{\text{SERR}}$ or $\overline{\text{PERR}}$) in an internal configuration register to allow the NMI handler software to determine the cause of the error.

The CY82C693 supports the four PCI interrupt signals ($\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, and $\overline{\text{INTD}}$). The interrupt lines are open-drain and should be routed to all of the PCI slots. For single-function devices, only $\overline{\text{INTA}}$ should be used. The three other interrupt lines can be connected to any set of functions on a multi-function device. Each interrupt signal can be programmed to be level-sensitive (PCI Compliant) or edge-triggered (not PCI Compliant). The assertion of a PCI interrupt request will cause the INTR signal to be asserted to the processor. When the processor performs an interrupt acknowledge cycle, the CY82C693 will return an interrupt vector based on the level of the PCI interrupt. The PCI interrupt levels are programmable. Interrupt programmability is useful in resolving system conflicts.

The CY82C693 contains the PCI arbiter. The CY82C693 supports up to five PCI masters, including the CY82C691. There are four dedicated request and grant line pairs, one for each slot, and a special busy and grant for the CY82C691. The CY82C691 is the default owner of the PCI bus. If the CY82C691 requires ownership of PCI and is granted the bus, it will assert the $\overline{\text{691BSY}}$ signal. When the CY82C693 sees another master's request asserted, and no higher priority masters are requesting the bus, the CY82C693 will remove the grant from the CY82C691 and give the grant to the new master. The priority is rotating. In rotating priority, the last master to be granted the bus becomes lowest priority. The CY82C691 is always given highest priority to reduce CPU latency due to arbitration.

ISA Bus Interface

The CY82C693 contains an ISA (Industry Standard Architecture) Bus interface. If no other target in the system claims the transaction, it is passed to the ISA bus. The ISA bus interface in the CY82C693 is full master/slave, allowing a myriad of low-cost peripheral cards to be added to the system.

ISA master cycles are performed by the CY82C693 whenever a PCI master wants to access an ISA peripheral card. The CY82C693 drives the address onto the ISA address lines and asserts BALE. The accessed resource will then respond with its size by the assertion/negation of the $\overline{\text{IOCS16}}$ or $\overline{\text{MEMCS16}}$ signals. The appropriate command signals ($\overline{\text{SMEMR}}$, $\overline{\text{SMEMW}}$, $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$, or $\overline{\text{IOW}}$) will then be asserted by the CY82C693. The de-assertion of the active command signal indicates when valid data is on the bus. Wait states can be inserted into an ISA cycle by the target's negation of the IOCHRDY signal. This forces the command signal to stay on the bus until IOCHRDY is asserted again.

An ISA peripheral requests ownership of the ISA bus by asserting its DMA request signal. The request can be either a normal ISA transaction controlled by the peripheral itself, or a DMA transaction using transfer parameters (such as starting address and block count) previously set-up in the DMA controller inside the CY82C693.

In a normal ISA transaction, the ISA master begins its transfers after receiving DMA acknowledge for the CY82C693. The CY82C693 differentiates a normal ISA bus request for an ISA DMA report by checking the transfer parameters (in the on-chip DMA controller) corresponding to the above DMA request signal. If no transfer parameters are stored, then the request is a normal ISA transfer.

If the ISA request is a DMA transfer, the CY82C693 outputs the starting address and the control signals onto the ISA bus. It will also deassert AEN to the requesting ISA bus master so that it will ignore address and simply provide the data to be transferred onto the ISA bus. If the target for a DMA/MASTER ISA cycle is on the PCI bus, the PCI bus will be arbitrated for before the CY82C693 will assert the DMA acknowledge to the ISA master.

Because a PCI data transfer is up to 32 bits wide on byte boundaries and ISA data can be eight or sixteen bits wide, the CY82C693 performs bus steering. Based on the address, the state of the C/$\overline{\text{BE}}$[3:0] signals and the width of the ISA bus resident, the CY82C693 will steer data to/from the PCI data bus from/to the ISA data bus. If a PCI read transaction from the ISA bus requires data larger than the 8/16-bit ISA data width, the CY82C693 can be programmed to perform multiple ISA transactions and pack the data into a single, larger PCI data word. If less than 32 bits are required on the PCI bus (by the assertion of fewer than all 4 byte enables), only the requested data will be packed and presented on the PCI data lines. On an ISA MASTER write, data will be driven onto the PCI bus as it is written (not packed).

The ISA bus interface in the CY82C693 allows for fully synchronous/asynchronous operation. In other words, the ISA clock can be synchronized to a divided-down version of the PCI clock, or the PCI and ISA clocks can be fully independent in frequency and phase. The CY82C693 will automatically switch the ISA clock to a fixed 7.16-MHz clock if the PCI clock is slowed to conserve system power.

The COMMAND recovery time is programmable (either 1.5 or 2.5 ISA clock cycles). Command recovery is th amount of idle ISA cycles between back-to-back ISA transactions. This allows for a trade-off between extra performance and card compatibility. The default is 2.5 ISA clock cycles. However, if the ISA peripheral cards in the system can support 1.5 ISA clocks, the CY82C693 can be programmed to reduce the overall ISA cycle time.

The CY82C693 contains QuietBus logic. Control and data signals will only be passed from PCI to ISA when the CY82C693 determines that the ISA bus is the intended target. This reduces the noise and power consumption associated with switching output buffers unnecessarily. It also eliminates some system EMI (Electro-Magnetic Interference).

Reset Logic

The CY82C693 provides three signals for resetting system components, CPURST, INIT and $\overline{\text{RESET}}$.

CPURST is an active high signal that provides power-on reset functionality for the CPU. CPURST is asserted when the CY82C693 detects PWGD (power good) asserted from the power supply. CPURST forces the processor to begin execution in a known state. When the Pentium processor detects CPURST, it will immediately abort all bus activity and perform its reset sequence. The CY82C693 will assert CPURST for a minimum of 1 ms after PWGD is asserted. This is sufficient to ensure a "cold" or "power-on" reset. The assertion of CPURST will cause all internal processor registers, write-buffers, and caches to be reset. The processor will begin execution by reading from address FFFFFFF0H upon the deassertion of CPURST. CPURST is also used to reset ISA peripherals (functions as SYSRESET).

For a "warm" reset, INIT will be asserted for a minimum of 15 CPU clock cycles. A "warm" reset is performed whenever the CPU writes Port B (61H), bit 0, to "1". The keyboard controller within the CY82C693 can also issue a "warm" reset if the user unites FE (hex) to port 64 (hex). INIT provides CPURST functionality except INIT leaves the CPU's level 1 cache, internal write-buffers, and floating-point registers intact. Only the processor core is reset. INIT can be used to switch the processor from protected to real mode. Once INIT is sampled active, the processor will begin the initialization sequence on the next instruction boundary. The initialization sequence will continue to completion followed by normal reset execution (read from address FFFFFFF0H). INIT will be asserted for a minimum of two CPU clock cycles and will remain active for three CPU clock cycles prior to the $\overline{\text{BRDY}}$ of an I/O write cycle.

$\overline{\text{RESET}}$ is used to reset all PCI peripherals (functions as $\overline{\text{PCIRST}}$). $\overline{\text{RESET}}$ will be asserted for a minimum of one PCI clock cycle.

Keyboard Controller

The CY82C693 integrates the essential functions of an 8042 Keyboard Controller including:

— operating frequencies from 6 to 16 MHz

— support for a PS/2-compatible mouse

— complete operating system independence

— works with MS-DOS®, Microsoft Windows®, OS/2®, and UNIX™

Operating Frequency

The keyboard controller inside the CY82C693 operates at frequencies between 6 and 16 MHz. The clock is internally selectable.

Resetting the Keyboard Controller

The keyboard controller will be reset when the PWGD (power good) signal is negated. Once PWGD is asserted, the keyboard controller will remain in its reset state for 120 keyboard clock cycles before becoming operational.

Host Interface

PCI or ISA masters communicate with the keyboard controller by performing I/O reads and writes to two eight-bit port locations (0064H and 0060H). I/O Port 0064H is the command/status register and I/O Port 0060H is the data register. There are two host signals that the keyboard controller generates, GTA20 and INIT. These signals initiate the A20 mask and ”warm reset” respectively. There are also six general purpose I/O registers and ports that are user defineable. As inputs, they set bits in internal reserved registers. As outputs, they reflect the value of the internal register bits. The general purpose I/Os are multiplexed with some alternate control signals (such as external RTCAS and Power Control Latch Enable), the XD bus, IDE DMA control, and IDE Interrupt Request signals. Therefore, these system functions will not be available from the CY82C693 when the pins are programmed as general purpose I/Os. However, not every system will require the additional system functions, and those that do can implement these system functions externally if general purpose I/O is a requirement.

PS/2 Compatible Mouse Support

The CY82C693 will support a PS/2 compatible mouse. MSCLK and MSDATA should be connected directly to the mouse connector. Mouse interrupt requests are generated internally.

Keyboard Interface

The CY82C693 provides the KBCLK and KBDATA signals to connect directly to the keyboard controller. There is also a KEYLOCK pin which should be connected to the KEYLOCK connector. If KEYLOCK is active, the keyboard controller will not respond to inputs from the keyboard. The keyboard interrupt request is internally connected to the CY82C693’s interrupt controller.

Maximum Flexibility

The internal keyboard controller can be disabled if a custom, external keyboard controller solution is desired. All of the signals needed to control and interface to an external keyboard controller are multiplexed with existing keyboard interface signals.

Power Management Logic

The CY82C693 provides flexible power management to help systems conform to governmental system power consumption guidelines. There are 5 timers within the power management logic (a standby timer, a suspend timer, user timer 1, user timer 2, and user timer 3). There are also 10 programmable event detectors.

Events which can be monitored include:

Keyboard Commands

Serial Port Commands

Parallel Port Commands

Hard Disk Commands

DMA/ISA MASTER Requests

A Specific (set of) Interrupt Request(s)

Video Memory Accesses

Floppy Drive Accesses

A PCI Master Request

A Specific I/O Range

The CY82C693 can monitor any combination or all of the above events.

Hardware power management can be selected to ease the power-down software requirements. In the hardware power management mode, the PMWTCNT signal and the $\overline{\text{STOPCLK}}$ signal may be programmed to automatically assert when the suspend and/or standby timers expire. These signals are used to reduce the power consumption of the processors and peripherals when they are idle.

Software power management works with CPU SMM (System Management Mode). All Pentium-class processors have SMM capabilities. SMM is entered through a dedicated System Management Interrupt ($\overline{\text{SMI}}$). SMM has its own protected address space, which can only be accessed in System Management Mode. All power management software should be embedded in the SMI handler code stored in SMM space. SMM memory control is handled by the CY82C691 (please see CY82C691 datasheet). However, all $\overline{\text{SMI}}$ generation is handled by the CY82C693.

The CY82C693 supports Full-Speed, Standby, and Suspend power-down states. Other user-defined power-down states can be implemented. Full-Speed is the normal operating state. When power-management is enabled, the Standby timer will begin counting. The terminal count for the Standby timer can be set to various values between 0.2 seconds and 240 minutes. If any of the monitored events occur before the terminal count is reached, the standby timer will reset to zero and begin counting again. If the standby timer expires without detecting any monitored events, the CY82C693 will assert $\overline{\text{SMI}}$, allowing the $\overline{\text{SMI}}$ handler to transition the system into the Standby state. The $\overline{\text{SMI}}$ signal must be cleared (deasserted) by the $\overline{\text{SMI}}$ handler. Once Standby state has been entered, the Suspend timer will begin counting. The terminal counts for the suspend timer can be set to values between 1 second and 480 minutes. If a Standby event is detected by the CY82C693 before the suspend timer expires, the CY82C693 will assert $\overline{\text{SMI}}$, reset all timers (only the Suspend timer will be reset if the event is a Suspend event), and set a status register bit to identify the monitored event. The System Management Interrupt handler must determine the source of the $\overline{\text{SMI}}$ by reading CY82C693 status registers. The handler code can use the $\overline{\text{SMI}}$ source information to determine which state to transition to. If the suspend timer expires before any monitored event occurs, the

CY82C693 will once again assert SMI so that the handler can transition to the Suspend state

The user-definable timers (User timer 1, User timer 2, and User timer 3) can be used to create additional power-down states. They can be enabled or disabled at any time. The user timer terminal count values can be programmed to values between 1 second and 480 minutes. When any of the timers expire, the CY82C693 will assert SMI (until it is cleared in the handler) and set a configuration register bit to indicate which timer(s) expired. The System Management Interrupt handler must read status registers to determine the source of the SMI. Software can then transition to any power-down state (pre-defined or user-defined). Only User timer 1 is programmable to be reset upon the detection of monitored events.

There are three power control signals used by the CY82C693 (PMWTCNT, STOPCLK, and EPMI). The first two signals (PMWTCNT and STOPCLK) are register controlled and can be asserted or negated in any state. PMWTCNT can be used as a latch enable to send a power-down code (driven on SD[7:0]) to a peripheral card. STOPCLK has been traditionally connected to the CPU's STOPCLK input to disable the clock to the internal CPU's core. When STOPCLK is enabled, it will periodically be asserted and deasserted based on programmable STOPCLK period registers. This signal may also be used to power-down peripherals that support a power-down signal. External logic can force the CY82C693 to assert SMI by activating the EPMI input.

The CY82C693 contains an internal interrupt/event counter. The terminal count for this timer should be programmed to the longest amount of time it takes to handle any interrupt. Whenever a user-selectable event (typically an interrupt) occurs, the STOPCLK signal will be deasserted for the time period of the interrupt timer. This allows the processor to handle the interrupt. If no other monitored events occur before the interrupt timer expires, STOPCLK will be reasserted.

AT Refresh Logic

The CY82C693 contains logic to support refresh cycles on the ISA bus. An internal refresh request is generated every 15.6 microseconds. Upon detecting the refresh request, the CY82C693 will arbitrate for the ISA bus and generate a refresh cycle. The CY82C693 contains its own internal refresh counter and refresh address counter. During ISA refresh cycles, the CY82C693 will not grant the ISA bus to any peripheral cards. If a PCI transaction is targeting ISA, the CY82C693 will attempt to buffer the transaction size of the buffer. If the internal FIFO buffers are full or the transaction cannot be buffered, the CY82C693 will negate TRDY until the refresh is completed.

Pre-Read/Post-Write Buffers

There are 3 FIFOs inside the CY82C693. PCI/ISA FIFOs are 2 deep by 32. There is one 4 deep by 32 bit FIFO for each IDE channel. PCI to ISA or IDE transactions are buffered to improve system performance. ISA or IDE transactions will likewise be buffered for transmission to the PCI bus. PIO IDE transactions go through their own independent buffers. The buffers eliminate some of the latency due to arbitration for resources.

BIOS ROM Control

The CY82C693 provides all of the control signals to support both Flash and conventional ROM. There is also a dedicated XD bus to buffer the ROM data (provided IDE DMA, Independent IDE Interrupt Requests, and several of the general purpose I/O functions are not required). The XD bus can also be externally buffered, which frees pins for other functions. If Flash ROM is used, the CY82C693 will provide the write-protection through an internal Read Only register bit.

Timer/Counter Logic

The CY82C693 contains an 8254-compatible timer/counter. It can be used to generate software time delays, count in binary or BCD, generate interrupts, or generate square-wave patterns. There are three independent, 16-bit counters that can operate in the following six modes: Interrupt on terminal count, Hardware retriggerable one-shot, Rate generator, Square wave generator, Software triggered strobe, and Hardware retriggerable strobe. The output of Counter 0 is internal only (can be programmed to generate interrupts). The output of Counter 1 is also internal and works with the AT refresh logic to generate ISA refresh cycles. The output of Counter 2 comes out on the SPKR pin and can be used in any of the modes for a variety of functions, including generating a square wave to an external speaker.

DMA Controllers

The CY82C693 contains two 8237 compatible DMA controllers cascaded together to provide seven separate DMA channels. Internally, each controller is a 4-channel DMA device which generates the memory address and control signals necessary to transfer data between an ISA peripheral device and system memory (via the PCI bus). The two DMA controllers (DMACs) are configured to provide four 8-bit DMA channels and three 16-bit DMA channels. Channel 0 of the 16-bit DMAC is used to cascade the two contollers. The DMA controllers support type A, B, and F transfer rates.

The DMACs operate in one of three states at all times: IDLE, PROGRAM, or ACTIVE.

After the DMACs have been initialized, they remain in IDLE until a DMA request signal is asserted. Once a DMA request is seen, the DMAC that receives the request will enter the ACTIVE state, or if the system is programming the internal configuration registers of a DMAC, it will enter the PROGRAM state. In the IDLE state, the DMACs do nothing aside from sampling the DMA request signals and configuration register decode signals. If a request comes in at the same time as a configuration register access, the register access is executed first.

The first thing the DMACs do, after entering the ACTIVE state, is arbitrate for the PCI bus. The CY82C693 must obtain ownership of the PCI bus regardless of whether the target is on PCI or not. After the PCI bus has been won, the CY82C693 will issue DMA acknowledge to the highest priority requestor. The DMA transfer is then free to begin. The DMAC is in control of the ISA command signals and IOCHRDY during DMA.

The DMACs will enter the PROGRAM state any time a PCI address is decoded to I/O Port 22H. This is the Index Register Port (IRP) for DMAC configuration space. A read or write to Port 23H should immediately follow the write to the IRP. Port 23H is the Data Register Port (DRP).

DMA Controller Transfer Modes

The DMACs can be programmed (on a per channel basis) to transfer data in four modes: Single Transfer Mode, Block Transfer Mode, Demand Transfer Mode, or Cascade Mode.

In Single Transfer Mode, the DMACs will transfer one byte/half-word of data at a time. A DMAC must arbitrate for the PCI bus after each transfer. This ensures that the PCI bus bandwidth will not be completely consumed by an ISA peripheral. The Word Count Register (WCR) will decrement from the block count until zero is reached. If autoinitializing is selected, the channel will reinitialize itself for the next DMA transfer. Otherwise, the channel will be masked until it is initialized.

In Block Transfer Mode, the DMACs will hold onto ownership of the PCI bus and continue performing transfers until the WCR is decremented to zero. This will give the best DMA performance but runs the risk of degrading overall system performance by tying up the PCI bus for long periods. The channel must once again be initialized/autoinitialized after the block transfer has been completed.

In Demand Transfer Mode, the DMACs will continue performing transfers until the WCR contains zero or the DMA request is negated. The peripheral will negate its DMA request when it is not ready to transfer data (e.g. an I/O device with its buffer full). When the device is ready to resume the transfer, it must assert its DMA request again and rearbitrate for the ISA and PCI busses. This allows higher priority bus masters to access the busses while the peripheral involved in the DMA cycle is not ready. However, if the device remains ready to transfer data, it can hold its DMA request active and perform back-to-back transfers without arbitrating between transfers. The channel must be initialized/autoinitialized after the block transfer has been completed.

Seven DMA channels are available in the CY82C693. If additional channels are required, channels may be placed in Cascade Mode. When a channel is in Cascade Mode, external 8237 HOLDREQ and HLDA signals can be routed to the channel's DMA request and DMA acknowledge signals respectively. The arbitration rotating protocol will be maintained.

Read, Write, and Verify transfers can be performed in any of the transfer modes. Any channel may also be configured to perform autoinitialization. During autoinitialization, the original values are automatically restored to the Base Address Register and Word Count Register from the Current Address Register and Current Word Count Register when the block transfer is complete (zero in the WCR or an EOP is signalled from the peripheral). The channel will not automatically be masked if autoinitialization is selected.

IDE Controller

The CY82C693 contains an integrated, dual-channel PCI to IDE bridge. The IDE controller conforms to ANSI modes 0, 1, 2, 3, and 4 for PIO (Programmed I/O) transfers. Single-word and multi-word DMA transfer modes 0, 1, and 2 are also supported. ATAPI (CD ROM) protocols are allowed with pre-fetching disabled. The controller allows for CHS (Cylinder-Head-Sector) or LBA (Logical Block Address) addressing. Each of the two channels support two devices for a maximum of four IDE devices in the system. The controller allows for PIO pre-fetching and DMA PCI mastering in order to increase overall system performance.

Real-Time-Clock

The CY82C693 contains a complete time-of-day clock with alarm, one hundred year calender, a programmable periodic interrupt generator, and 242 bytes of battery-backed "scratch" RAM. The entire Real-Time-Clock (RTC) remains operational under normal or battery power. A 32 kHz oscillator is integrated as part of the RTC. Only an external crystal and a battery are required to complete the clock circuit.

RTC Address Map

The internal RTC contains 256 bytes of battery-backed RAM (14 bytes for clock data and 242 bytes of user-definable RAM). *Figure 2* shows the address map.

	Hex Address
Seconds	00
Seconds Alarm	01
Minutes	02
Minutes Alarm	03
Hours	04
Hours Alarm	05
Day of the week	06
Numerical date of the month	07
Month	08
Year	09
Register A	0A
Register B	0B
Register C	0C
Register D	0D
114 Bytes of User-Defined RAM (Configuration Storage)	0E 7F
128 Bytes of additional User-Defined RAM (Configuration Storage)	F0 FF

Figure 2. Real-Time-Clock Address Map

Update cycles are performed once per second. The update cycle increments the seconds byte and increments the minutes byte on an overflow (followed by hour, day, month, year, etc.). Alarm value comparisons are also made. During the update cycle, data is undefined. However, the CY82C693 contains an UIP (update in progress) bit to inform the system of updates. If the processor, upon polling the appropriate RTC status register, sees the UIP bit set, an access to the RTC clock data should not be performed. An update-ended interrupt may also be programmed to inform the processor that the update cycle has ended and RTC data is valid.

External RTC Control

To allow for maximum flexibility, an external RTC can be used. The control signals for an external RTC are only used when the internal RTC is disabled.

Interrupt Controllers

The CY82C693 contains two interrupt controllers (INTC1 and INTC2) that provide 82C59A functionality. There are thirteen separate interrupt request inputs (although some of the interrupt requests are only available internally). The two controllers are cascaded to maintain AT interrupt priorities. Interrupt arbitration can be programmed to be rotating or fixed. When interrupt requests come in from the system, the CY82C693 will store all requests, evaluate the priority, and respond with the appropriate acknowledge vector for the CPU's first interrupt acknowledge cycle. Then, if automatic end-of-interrupt (AEOI) is selected, the internal interrupt pending bit will be reset on the second CPU interrupt acknowledge cycle. If AEOI is not selected, it is the responsibility of the software to generate the appropriate end-of-interrupt (EOI) sequence at the end of the interrupt service routine to clear down the interrupt (for example, if a clock interrupt is requested, the system designer may wish the pending interrupt request to remain active through the entire service routine. This allows higher priority interrupts to prematurely force an exit from the clock service routine without causing the loss of the clock interrupt request. If AEOI were programmed, the request to the CPU and the interrupt pending bit would automatically be reset before the clock service routine was entered. If a higher priority interrupt forced the clock routine to be exited, the clock interrupt would be lost.)

The interrupt controllers (INTCs) are initialized and programmed using special command words issued by the CPU. Initialization Command Words (ICW1 through ICW4) bring the INTCs to a known state when the system is first powered-up or reset. Operational Command Words (OCW1 through OCW3) setup the various operating modes.

The following events occur automatically in the initialization sequence:

— The edge sense circuit is reset. Therefore, a request must make a Low-to-High transition to be detected. If the interrupt request is programmed to be edge-detected, the rising edge must be used (Low-to-High). High-to-Low transitions will be ignored.

— The interrupt mask register is cleared. Interrupts are enabled.

— Interrupt Request (IR) 7 is set to the lowest priority on each INTC.

— Special Mask Mode is reset.

— Status Read is set to return the value in the Interrupt Request Register (IRR).

The following options are programmable through the ICWs and OCWs:

— Vector Mode operation: The CY82C693 can generate the vector for an interrupt acknowledge cycle.

— The call address interval: The number of cycles to generate an interrupt vector can be chosen to be 4 or 8.

— Edge vs. Level Sensitive Interrupt Requests: The interrupt request lines can be programmed to be detected on a rising edge or a fixed level on an individual request basis.

— Vector Address Byte: The value can be programmed.

— Automatic End-Of-Interrupt: The pending interrupt can be programmed to clear automatically with the acknowledge cycle. If AEOI is not programmed, interrupts must be cleared within the interrupt service routine by the software.

— Interrupt Nesting: Interrupt nesting is a programmable option. If interrupt nesting is allowed, interrupts can be asserted within interrupt service routines.

— Interrupt Masking: The bits of the Interrupt Mask Register can be set to mask (ignore) certain interrupt requests.

— Interrupt Priority: Priority can be programmed to be rotating or fixed.

— Polling: When Polling is selected, Interrupt Requests will not issue an interrupt to the CPU. They will merely set bits within the Interrupt Pending Register (IPR). It is the responsibility of the CPU to periodically read (poll) the IPR to determine if an interrupt is pending.

PCI and IDE interrupts are fully routable internally to independent interrupt levels. The interrupt level of each PCI or IDE interrupt is controlled with configuration registers.

CY82C693 Signal Description

The CY82C693 signals are divided into seven functional areas: Reset signals, PCI Interface signals, ISA Interface signals, Power Management signals, Keyboard Interface signals, IDE Interface signals, and Miscellaneous signals.

Reset Signals

Name	I/O	Description
PWGD	I	Power Good: This signal is driven active from a combination of the external power supply's power good signal and the external reset switch. This signal is used to qualify initialization signals and reset the internal state of the CY82C693.
CPURST	O	CPU Reset: This signal resets the CPU and the ISA bus.
$\overline{\text{RESET}}$	O	PCI Reset: This signal functions as $\overline{\text{PCIRST}}$ to reset the PCI bus.
INIT	O	CPU Initialization: This signal is used to reset the core of the CPU without disturbing the state of internal caches or write-buffers. This signals can be used to switch the processor from protected mode to real mode.

PCI Interface Signals

Name	I/O	Description
AD[31:0]	I/O	PCI Address/Data Bus: Multiplexed bidirectional address/data lines on the PCI bus. The CY82C693 either drives or samples these lines during PCI cycles.
PCICLK	I	PCI Clock: PCI Clock Input. This signal is used to synchronize the CY82C693 to the PCI bus. The clock must be within the range 25 MHz to 33 MHz.

PCI Interface Signals (continued)

Name	I/O	Description
C/$\overline{\text{BE}}$[3:0]	I/O	PCI Command & Byte Enables: C/$\overline{\text{BE}}$[3:0] are driven by the current bus master during the address phase to define the transaction and during the data phase as the byte enables. These signals are outputs when the CY82C693 is a master and inputs when the CY82C693 is a slave.
$\overline{\text{FRAME}}$	I/O	Cycle Frame: Driven by the current bus master to indicate the start and duration of a transaction.
$\overline{\text{IRDY}}$	I/O	Initiator Ready: The assertion of $\overline{\text{IRDY}}$ indicates the current bus master's ability to complete the current data phase of the transaction. Used in conjunction with $\overline{\text{TRDY}}$ from the target.
$\overline{\text{TRDY}}$	I/O	Target Ready: The assertion of $\overline{\text{TRDY}}$ indicates the current target's ability to complete the current data phase of the transaction. Works in conjunction with $\overline{\text{IRDY}}$ from the master.
$\overline{\text{DEVSEL}}$	I/O	Device Select: Indicates that a PCI device has decoded that it is the target of the transaction. The target has three options for decoding: fast decoding, medium decoding, or slow decoding. The CY82C693 will sample $\overline{\text{DEVSEL}}$, and if it is not asserted by a target within the timeout period, will assert $\overline{\text{DEVSEL}}$ to claim the cycle.
PAR	I/O	Parity: An even parity bit across AD[31:0] and C/$\overline{\text{BE}}$[3:0]. As a master the CY82C691 generates even parity on PCI write cycles. On read cycles, the CY82C693 checks parity by sampling PAR.
$\overline{\text{STOP}}$	I/O	Stop: Indicates that the current target is requesting the master to stop the current transaction. $\overline{\text{STOP}}$ is used in conjunction with $\overline{\text{DEVSEL}}$ and $\overline{\text{TRDY}}$ to indicate disconnect, target abort, and retry cycles.
$\overline{\text{LOCK}}$	I	PCI Lock: Used to indicate that an atomic operation is taking place and may require multiple cycles to complete without another master interfering. The CY82C693 will accept atomic transactions but never issues them.
$\overline{\text{PERR}}$	I/O	Parity Error: Parity Error may be asserted by any agent that detects a parity error during the data phase of a transaction. Address parity errors are reported on $\overline{\text{SERR}}$. $\overline{\text{PERR}}$ will cause the CY82C693 to assert NMI to the processor.
$\overline{\text{SERR}}$	I/O	System Error: System error may be asserted by any agent for reporting address parity errors or any other types of errors beside data parity. $\overline{\text{SERR}}$ will cause the CY82C693 to assert NMI to the processor.
IDSEL	I	PCI ID Select: This signal should be connected to a unique PCI address. It is used to select the CY82C693 during PCI configuration cycles.
$\overline{\text{INTA/B/C/D}}$	I	PCI Interrupt Requests: These signals allow PCI peripherals to interrupt the processor.
$\overline{\text{691BSY}}$	I	CY82C691 Busy: The CY82C691 asserts busy to indicate to the central arbiter in the CY82C693 that the CY82C691 owns the PCI bus. When not asserted, the CY82C691 is not ready to own the PCI bus or does not require PCI bus ownership.
$\overline{\text{691GNT}}$	O	CY82C691 Grant: When asserted, it indicates to the CY82C691 that it has been granted use of the PCI bus and is allowed to initiate a transaction. This is the highest priority PCI grant signal. The PCI arbitration will be parked on this grant.
$\overline{\text{REQ}}$[3:0]	I	PCI Bus Requests: These signals are connected to the individual bus requests from each PCI peripheral. When a combination of the bus requests is asserted, the CY82C693 will resolve the priority and give the grant to the highest priority master.
$\overline{\text{GNT}}$[3:0]	O	PCI Bus Grants: These signals are connected to the individual bus grants of each PCI peripheral. When a combination of the bus requests is asserted, the CY82C693 will resolve the priority and assert grant to the highest priority master.

ISA Interface Signals

Name	I/O	Description
SA[19:4]/ PCIIDE[15:0]	I/O	ISA Address Bus/PCI IDE Data Bus: These signals provide most of the address for the ISA bus as well as the data path for IDE.
SA[3:0]	I/O	ISA Address: These signals provide the rest of the ISA address.
SD[15:0]	I/O	System Data Bus: These signals connect directly to the ISA data bus.
XD7/PIR0 GPIO0	I/O	XD Bus, Bit 7/Programmable Interrupt Request 0/General Purpose I/O 0: If zero TTL is desired for XD bus support (BIOS ROM data, external keyboard controller data, and external RTC data), this is bit 7 of the data bus. Otherwise, this is user-selectable to provide support for a Programmable Interrupt Request or a general purpose I/O. As a GPIO, this signal can either reflect the contents of an internal register bit (output) or set an internal register bit to the value driven on the line (input).
XD6/PIR1/ GPIO1	I/O	XD Bus, Bit 6/Programmable Interrupt Request 1/General Purpose I/O 1: If zero TTL is desired for XD bus support (BIOS ROM data, external keyboard controller data, and external RTC data), this is bit 6 of the data bus. Otherwise, this is user-selectable to provide support for a Programmable Interrupt Request or a general purpose I/O. As a GPIO, this signal can either reflect the contents of an internal register bit (output) or set an internal register bit to the value driven on the line (input).
XD5/IDEDREQ0/ GPIO2	I/O	XD Bus, Bit 5/IDE DMA Request 0/General Purpose I/O 2: If zero TTL is desired for XD bus support (BIOS ROM data, external keyboard controller data, and external RTC data), this is bit 5 of the data bus. Otherwise, this is user-selectable to provide support for IDE DMA or a general purpose I/O. For IDE DMA support, this signal is DMA request 0 (primary IDE channel DMA request). As a GPIO, this signal can either reflect the contents of an internal register bit (output) or set an internal register bit to the value driven on the line (input).
XD4/$\overline{\text{IDEDACK0}}$/ GPIO3	I/O	XD Bus, Bit 4/IDE DMA Acknowledge 0/General Purpose I/O 3: If zero TTL is desired for XD bus support (BIOS ROM data, external keyboard controller data, and external RTC data), this is bit 4 of the data bus. Otherwise, this is user-selectable to provide support for IDE DMA or a general purpose I/O. For IDE DMA support, this signal is DMA acknowledge 0 (primary IDE channel DMA acknowledge) . As a GPIO, this signal can either reflect the contents of an internal register bit (output) or set an internal register bit to the value driven on the line (input).
XD3/IDEREQ1/ GPIO4	I/O	XD Bus, Bit 3/IDE DMA Request 1/General Purpose I/O 4: If zero TTL is desired for XD bus support (BIOS ROM data, external keyboard controller data, and external RTC data), this is bit 3 of the data bus. Otherwise, this is user-selectable to provide support for IDE DMA or a general purpose I/O. For IDE DMA support, this signal is DMA request 1 (secondary IDE channel DMA request). As a GPIO, this signal can either reflect the contents of an internal register bit (output) or set an internal register bit to the value driven on the line (input).
XD2/$\overline{\text{IDEDACK1}}$/ GPIO5	I/O	XD Bus, Bit 2/IDE DMA Acknowledge 1/General Purpose I/O 5: If zero TTL is desired for XD bus support (BIOS ROM data, external keyboard controller data, and external RTC data), this is bit 2 of the data bus. Otherwise, this is user-selectable to provide support for IDE DMA or a general purpose I/O. For IDE DMA support, this signal is DMA acknowledge 1 (secondary IDE channel DMA acknowledge) . As a GPIO, this signal can either reflect the contents of an internal register bit (output) or set an internal register bit to the value driven on the line (input).
XD1/$\overline{\text{XDEN}}$	I/O	XD Bus, Bit 1/External XD Bus Buffer Enable: If zero TTL is desired for XD bus support (BIOS ROM data, external keyboard controller data, and external RTC data), this is bit 1 of the data bus. Otherwise, this pin provides the enable for a buffer between XD[7:0] and SD[7:0].
XD0/XDIR	I/O	XD Bus, Bit 0/External XD Bus Direction Control: If zero TTL is desired for XD bus support (BIOS ROM data, external keyboard controller data, and external RTC data), this is bit 0 of the data bus. Otherwise, this pin provides the direction control for a buffer between XD[7:0] and SD[7:0].

ISA Interface Signals (continued)

LA21/$\overline{\mathrm{SIDECS1}}$	I/O	Latched Address 21/ Secondary IDE Chip Select 1: This signal connects to LA21 on the ISA bus and is used to provide access up to 16MB. During IDE accesses, this signal provides the chip select 1 for the secondary channel.
LA20/$\overline{\mathrm{SIDECS3}}$	I/O	Latched Address 20/ Secondary IDE Chip Select 3: This signal connects to LA20 on the ISA bus and is used to provide access up to 16MB. During IDE accesses, this signal provides the chip select 3 for the secondary channel.
LA[19:17]/ PCIIDEA[2:0]	I/O	Latched Address 19 Through 17/ PCI IDE Register Select Address 2 Through 0: These signals connect to LA19 through LA17 on the ISA bus and are used to provide access up to 16MB. During IDE accesses, these signals provide the register select (or address) to the IDE connectors.
KBCLK/IRQ1	I/O	Keyboard Clock/Interrupt Request 1: This signal is the keyboard clock connected to the keyboard connector if the internal keyboard controller is used. Otherwise, it provides IRQ1 (the keyboard controller interrupt input) if an external keyboard controller is desired.
IRQ[15:14], IRQ[11:9] IRQ[7:3]	I	Interrupt Request Inputs: These signals provide interrupt requests for CPU interrupters.
MSCLK/IRQ12	I/O	Mouse Clock/Interrupt Request 12: This signal is the mouse clock connected to the mouse connector if the internal keyboard controller is used. Otherwise, it provides IRQ12 (the mouse controller interrupt input) if an external keyboard controller is desired.
$\overline{\mathrm{PSRSTB}}$/$\overline{\mathrm{IRQ8}}$	I	Power Strobe/Interrupt Request 8: If the internal Real-Time-Clock (RTC) is used, this is the power strobe input. If an external RTC is desired, this is interrupt request 8 (traditionally the RTC interrupt).
INTR/$\overline{\mathrm{PPCIMST}}$	I/O	CPU Interrupt/Primary IDE Master Enable: This is the INTR (interrupt request) signal that is connected directly to the CPU's INTR pin. When interrupt requests come in to the CY82C693, it will assert INTR to the CPU. On power-up this signal should be pulled-down through a 1K Ohm resistor to inform the CY82C693 that the primary IDE channel requires DMA master support.
NMI/$\overline{\mathrm{SPCIMST}}$	I/O	Non-maskable CPU Interrupt/Secondary IDE Master Enable: This is the NMI (non-maskable interrupt request) signal that is connected directly to the CPU's NMI pin. When the CY82C693 detects a fatal error (such as a PCI parity error), it will assert NMI to the CPU. On power-up this signal should be pulled-down through a 1K Ohm resistor to inform the CY82C693 that the secondary IDE channel requires DMA master support.
ATCLK	O	AT Clock: This signal can be used to provide the system ISA (AT) bus clock. It can be internally programmed to generate different ISA frequencies.
BALE	I/O	Bus Address Latch Enable: This signal is the ISA BALE signal used by peripherals to latch the cycle address, AEN, and $\overline{\mathrm{SBHE}}$.
DREQ[7:5], DREQ[3:0]	I	ISA DMA/Master Request Inputs: These signals are used by external ISA peripherals to request mastership of the ISA bus for DMA or ISA MASTER cycles.
$\overline{\mathrm{DACK0}}$/TESTM	I/O	ISA DMA/Master 0 Acknowledge/Test Mode Enable: This signal is used by the CY82C693 to grant ISA bus mastership to external ISA peripherals for DMA or ISA MASTER cycles. At power-up, this pin should be pulled-down through a 1K Ohm resistor to enable test mode. For normal operation, this pin should not be pulled-down.
LA23/$\overline{\mathrm{IDECS1}}$	I/O	Latched Address 23/ IDE Chip Select 1: This signal connects to LA23 on the ISA bus and is used to provide access up to 16MB. During IDE accesses, this signal provides the chip select 1 for the primary channel.
LA22/$\overline{\mathrm{IDECS3}}$	I/O	Latched Address 22/ IDE Chip Select 3: This signal connects to LA22 on the ISA bus and is used to provide access up to 16MB. During IDE accesses, this signal provides the chip select 3 for the primary channel.

ISA Interface Signals (continued)

Name	I/O	Description
$\overline{\text{DACK1}}$/TESTM0	I/O	ISA DMA/Master 1 Acknowledge Input/Test Mode Select 0: This signal is used by the CY82C693 to grant ISA bus mastership to external ISA peripherals for DMA or ISA MASTER cycles. At power-up, this pin should be pulled-down through a 1K Ohm resistor or not pulled-down to select between the different test modes.
$\overline{\text{DACK2}}$/TESTM1	I/O	ISA DMA/Master 2 Acknowledge/Test Mode Select 1: This signal is used by the CY82C693 to grant ISA bus mastership to external ISA peripherals for DMA or ISA MASTER cycles. At power-up, this pin should be pulled-down through a 1K Ohm resistor not pulled-down to select between the different test modes.
$\overline{\text{DACK3}}$/DISPSEL	I/O	ISA DMA/Master 3 Acknowledge Input/Display Type Select: This signal is used by the CY82C693 to grant ISA bus mastership to external ISA peripherals for DMA or ISA MASTER cycles. At power-up, this pin should be pulled-down through a 1K Ohm resistor or left floating to select between CGA and Monochrome monitors (Acts as the keyboard controller Mono/Color pin).
$\overline{\text{DACK5}}$/KBSEL	I/O	ISA DMA/Master 5 Acknowledge Input/Internal Keyboard Controller Enable: This signal is used by the CY82C693 to grant ISA bus mastership to external ISA peripherals for DMA or ISA MASTER cycles. At power-up, this pin should be pulled-down through a 1K Ohm resistor to disable the internal keyboard controller if an external keyboard controller is desired.
$\overline{\text{DACK6}}$/RTCSEL	I/O	ISA DMA/Master 6 Acknowledge Input/Internal Real Time Clock Enable: This signal is used by the CY82C693 to grant ISA bus mastership to external ISA peripherals for DMA or ISA MASTER cycles. At power-up, this pin should be pulled-down through a 1K Ohm resistor to disable the internal RTC if an external RTC is desired.
$\overline{\text{DACK7}}$/IDESEL	I/O	ISA DMA/Master 7 Acknowleged Input/Internal IDE Controller Enable: This signal is used by the CY82C693 to grant ISA bus mastership to external ISA peripherals for DMA or ISA MASTER cycles. At power-up, this pin should be pulled-down through a 1K Ohm resistor to disable the internal IDE controller if an external IDE controller is desired.
$\overline{\text{REFSH}}$	I/O	AT Refresh: Driven when an ISA refresh cycle is in progress. As an input, this signal will force a refresh cycle to be initiated.
SPKR	O	Speaker Output: This signal is the output of counter 2 in the timer/counter logic. It can be used to drive a speaker.
EOP	O	End of Process: This signal is driven by the CY82C693 to signal the end of a block transfer during a DMA cycle.
$\overline{\text{IGNNE}}$	O	Ignore Numerical Error: This signal is driven by the CY82C693 to the CPU and should be connected to the CPU’s $\overline{\text{IGNNE}}$ pin. When this signal is asserted, the processor will ignore numerical errors and continue executing non-control, floating-point instructions.
$\overline{\text{FERR}}$	O	Floating Point Error: This signal informs the processor that a coprocessor error has occurred.
GTA20	I/O	Gate A20: This signal forces memory to wrap-around 1 MB. It is implemented as a fast gate A20.
RTCAS/GPIO7	I/O	RTC Address Strobe/General Purpose I/O 7: This signal provides the address strobe for an external Real-Time Clock. If the internal RTC is used, this signal acts as a GPIO. This signal can either reflect the contents of an internal register bit (output) or set an internal register bit to the value driven on the line (input).
OSCO/$\overline{\text{RTCRD}}$	O	RTC Oscillator Out/RTC Output Enable: If the integrated RTC is used, this signal is the 32.768 Khz crystal output. If an external RTC is desired, this signal is the RTC output enable.
OSCI/$\overline{\text{RTCWT}}$	I/O	RTC Oscillator In/RTC Write Enable: If the integrated RTC is used, this signal is the 32.768 kHz crystal input. If an external RTC is desired, this signal is the RTC write enable.
$\overline{\text{IOCS16}}$	I	16-Bit I/O Chip Select: This signal is driven active by any ISA I/O target that can support 16-bit accesses.

ISA Interface Signals (continued)

Name	I/O	Description
$\overline{\text{MCS16}}$	I	16-Bit Memory Chip Select: This signal is driven active by any ISA MEMORY target that can support 16-bit accesses.
$\overline{\text{SBHE}}$	I/O	System Byte High Enable: This signal is driven active by the current ISA bus master to indicate that valid data resides on SD[15:8].
$\overline{\text{MEMR}}$	I/O	ISA Memory Read Command Signal: This signal is driven by the current ISA bus owner to request a memory resource to drive data onto the bus during a cycle.
$\overline{\text{MEMW}}$	I/O	ISA Memory Write Command Signal: This signal is driven by the current ISA bus owner to request a memory resource to accept data presented on the bus during a cycle.
$\overline{\text{IOR}}$	I/O	ISA I/O Read Command Signal: This signal is driven by the current ISA bus owner to request an I/O resource to drive data onto the bus during a cycle.
$\overline{\text{IOW}}$	I/O	ISA I/O Write Command Signal: This signal is driven by the current ISA bus owner to request an I/O resource to accept data presented on the bus during a cycle.
$\overline{\text{SMEMR}}$	I/O	ISA System Memory Read Command Signal: This signal is driven by the current ISA bus owner to request a memory resource to drive data onto the bus during a cycle. This signal is only active within the first 1MB of memory space.
$\overline{\text{SMEMW}}$	I/O	ISA System Memory Write Command Signal: This signal is driven by the current ISA bus owner to request a memory resource to accept data presented on the bus during a cycle. This signal is only active within the first 1MB of memory space.
OSC	I	Oscillator Input: This signal is the 14.318 MHz reference clock input, used by the timer and DMA controller.
$\overline{\text{0WS}}$	I	Zero Wait State Input: When this signal is driven by an ISA peripheral, the CY82C693 will shorten the ISA cycle to zero wait states and ignore IOCHRDY.
AEN	O	Address Enable: This signal is driven when a valid address is present on the ISA bus. It is deasserted during DMA cycles so that ISA peripherals do not respond to the DMA memory cycle.
IOCHRDY	I/O	I/O Channel Ready: When deasserted, this signal indicates to the ISA bus owner that additional cycle time is required to complete the transaction. Used to add ISA wait states.
$\overline{\text{IOCHK}}$	I	I/O Channel Check: This signal is driven active if the system detects an ISA bus parity error.

Power Management Signals

Name	I/O	Description
$\overline{\text{SMI}}$	O	System Management Interrupt: Used by the CY82C693 to send the highest-level, process-transparent interrupt to the CPU.
$\overline{\text{STOPCLK}}$	O	CPU Stop Clock: This signal is used to stop the clock to the CPU's core to reduce power-consumption during idle periods.
$\overline{\text{EPMI}}$	I	External Power Management Input: This signal is used to force the assertion of an $\overline{\text{SMI}}$ by an external source (e.g., a power-down switch).
PMWTCNT/ GPIO6	I/O	Power Management Write Control/General Purpose I/O 6: This signal can be used as a control signal to enable an external power control latch. The latch can send peripheral power-down code sequences. As a GPIO, this signal can either reflect the contents of an internal register bit (output) or set an internal register bit to the value driven on the line (input). $\overline{\text{SLOWCLK}}$ functionality can also be implemented.

Keyboard Interface Signals

Name	I/O	Description
KBDATA	I/O	Keyboard Data: This is the serial data line to/from the keyboard connector.
$\overline{\text{KBCS}}$/KEYLOCK	I/O	Keyboard Chip Select/Key Lock Input: This signal is the key lock input to lock the keyboard (for system security) if the internal keyboard controller is used. If an external keyboard controller is desired, this is the keyboard controller chip select.
MSDATA	I/O	Mouse Data: This is the serial mouse data line to/from the keyboard connector.

IDE Interface Signals

Name	I/O	Description
$\overline{\text{IDEIOR}}$	O	IDE I/O Read: This signal directly drives the $\overline{\text{IDEIOR}}$ signal on the IDE connector.
$\overline{\text{IDEIOW}}$	O	IDE I/O Write: This signal directly drives the $\overline{\text{IDEIOW}}$ signal on the IDE connector.
$\overline{\text{IDEIOCS16}}$	O	IDE I/O Chip Select 16: This signal directly drives the $\overline{\text{IDEIOCS16}}$ signal on the IDE connector. Determines whether the IDE transaction is 8 bits or 16 bits wide.

Miscellaneous Signals

Name	I/O	Description
$\overline{\text{ROMCS}}$	O	System ROM Chip Select: This signal is used to enable the on-board BIOS ROM.
VBATT	I	Battery Backup Power: This pin should be connected directly to a battery-driven power source. Used to retain and maintain RTC functionality when the main power supply is disconnected.
+5V	I	V_{CC}: These are the +5V power supply pins for the device. They should be maintained within the proper operating limits.
+3.3V	I	3.3-Volt V_{CC}: This is the +3.3V power supply pin for the device. It should be maintained within the proper operating limits.
GND	I	GROUND: These are the 0V power supply pins for the device. They should be maintained within the proper operating limits.

Maximum Ratings

(Above which the useful life may be impaired. For user guidelines, not tested.)

Supply Voltage (V_{CC}) .... +7 V
Ambient Operating Temperature .... −25°C to +70°C
Ambient Storage Temperature .... −40°C to 125°C
DC Voltage Applied to Outputs .... −0.5V to V_{DD}
DC Input Voltage .... −0.5V to V_{DD}

Electrical Characteristics Over the Operating Range (T_A=0°C to 70°C)

Parameter	Description		Min.	Max.	Unit
V_{CC}	Core Supply Voltage		4.5	5.5	V
V_{DD}	3.3V I/O Supply Voltage		3.0	Vcc	V
V_{IL}	Input LOW Voltage		−0.5	0.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{DD}+0.5	V
V_{OL}	Output LOW Voltage			0.4	V
V_{OH}	Output HIGH Voltage		2.4		V
I_{IL}	Input Leakage Current			10	μA
I_{OL}	Output Leakage			10	μA
C_{IN}	Input Capacitance			10	pF
C_{OUT}	Output Capacitance			10	pF
I_{CC}	Power Supply Current	66 MHz		TBD	mA

Document #: 38−00457