μPD43608
SINGLE-CHIP CACHE SUBSYSTEM

Description

The μPD43608 is an integrated cache subsystem that provides the microprocessor system designer with a high-performance, single-chip, general-purpose cache solution. The μPD43608 consists of a CPU interface, directory storage (including address tag and validity bit storage), 8K bytes of on-chip data storage, 128 x 6-bit least recently used (LRU) replacement storage, internal address and data paths for cache bypass operations, an asynchronous 32-bit system bus interface, and several features optimizing cache write and miss operations. The μPD43608 is also able to interface with a number of 16- and 32-bit general-purpose microprocessors operating at 16 or 20 MHz.

Features

- ☐ High-performance 16- and 20-MHz operation
- ☐ 16- and 32-bit microprocessor interface capability
- ☐ Integrated cache architecture
  - — 8K bytes of on-chip data storage
  - — 16-byte cache block size
  - — 4-way set associative placement algorithm
- ☐ Bus monitoring circuit
- ☐ LRU replacement algorithm
- ☐ Prefetch on miss—one block lookahead
- ☐ Fetch bypass and wraparound load
- ☐ Asynchronous 32-bit system bus interface
- ☐ Multichip configuration increases cache size
- ☐ Write-through storage update policy with one-level write buffer
- ☐ 132-pin ceramic pin grid array packaging
- ☐ CMOS circuit technology

Ordering Information

Part Number	Ready Output Time (max)	Cycle Time (min)	Package
μPD43608R-2	70 ns	125 ns	132-pin ceramic pin grid array
R-3	50 ns	100 ns	

Organization

The μPD43608 is organized as a 4-way set associative cache, with 8K bytes of on-chip data storage organized as 128 sets by four 16-byte data blocks. When the CPU executes a read cycle, the address tag field of the physical CPU address is compared to the address tag in the cache directory. If a hit occurs, the selected data is sent to the CPU. Otherwise, the μPD43608 initiates a miss cycle to access main storage and update the cache with the replacement block. This architecture ensures a high hit ratio of 95% in most microprocessor applications.

Optimizing the Miss Cycle

The hit rate is an important parameter for measuring performance. Since a high hit rate of 95% requires that the μPD43608 access the main storage array for 5% of all read cycles, the penalty in system performance incurred during a miss cycle may be significant. The μPD43608 provides a number of on-chip features that optimize system performance during a miss cycle.

Data Transfer Cycles

The μPD43608 cache subsystem provides two data transfer modes for accessing main storage during a miss cycle: (1) burst data transfer mode uses the nibble access feature of a DRAM in main storage to optimize system bus bandwidth; (2) in single data transfer mode, an address is transmitted with each read cycle to main storage for systems that don't use nibble access DRAMs.

Block Load and Fetch Bypass Buffers

Once the replacement block has been read from main storage, the block load buffer is used to reduce the replacement block transfer time by providing a temporary buffer for storing the replacement block while the cache data storage is being updated.

Contact your NEC sales representative for a copy of the complete data sheet.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Concurrently, the CPU throughput is optimized by loading the missed word into the fetch bypass buffer as soon as it is read from main storage. The CPU directly accesses the fetch bypass buffer and can fetch the missed word without having to wait for the replacement block to be stored in cache data storage. If the CPU attempts to read the next word in the replacement block, the cache searches the directory and the block load buffer to determine whether or not a hit has occurred. Once the entire replacement block is loaded into the block load buffer, the data is wraparound-loaded into cache data storage.

Prefetch on Miss

On cache miss cycles, the μPD43608 implements a one-block lookahead algorithm that prefetches the next sequential cache data block, thus increasing the cache hit rate. Although prefetching can improve cache performance, a check must be made to determine that the block is not currently stored in the cache. The μPD43608 performs this check during each prefetch cycle, searching the cache directory for the desired prefetch block. If a hit occurs, the prefetch logic aborts the cycle. This function, which ensures that the cache is not polluted with duplicate data, can be enabled or disabled by controlling the cache status code signals during each read cycle.

Replacement Algorithm

The μPD43608 uses a least recently used (LRU) replacement algorithm to determine which data block should be overwritten during a cache miss cycle. This algorithm improves cache performance by choosing the data block with the least usage to optimize the hit rate.

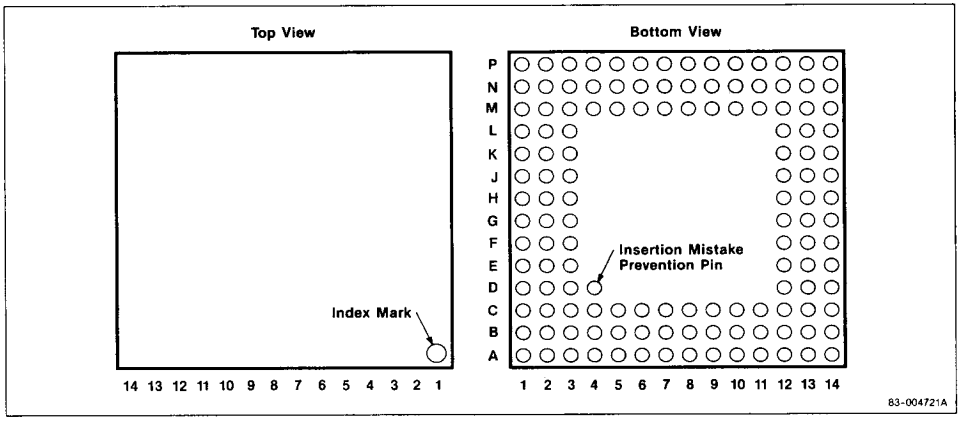
Main Storage Update Policies

To maintain data consistency in the storage hierarchy during each cache write cycle, the μPD43608 uses a write-through method that updates the main storage as soon as the CPU writes data to cache storage. CPU throughput is optimized by means of a one-level write buffer, which temporarily stores write data and initiates the write cycle to main storage, allowing the CPU to concurrently execute the next instruction.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Pin Configuration

132-Pin Ceramic Pin Grid Array



Pin Number	Function
A_1	D_{15}
A_2	D_{12}
A_3	D_{10}
A_4	D_9
A_5	D_7
A_6	D_5
A_7	D_3
A_8	D_2
A_9	A_1
A_{10}	A_2
A_{11}	A_5
A_{12}	A_7
A_{13}	A_{10}
A_{14}	A_{12}
B_1	D_{20}
B_2	D_{17}
B_3	D_{13}
B_4	D_{11}
B_5	D_8
B_6	D_6
B_7	D_4
B_8	D_1
B_9	D_0
B_{10}	A_3
B_{11}	A_6
B_{12}	A_9
B_{13}	A_{13}
B_{14}	A_{17}
C_1	D_{22}
C_2	D_{19}
C_3	D_{16}
C_4	D_{14}
C_5	GND

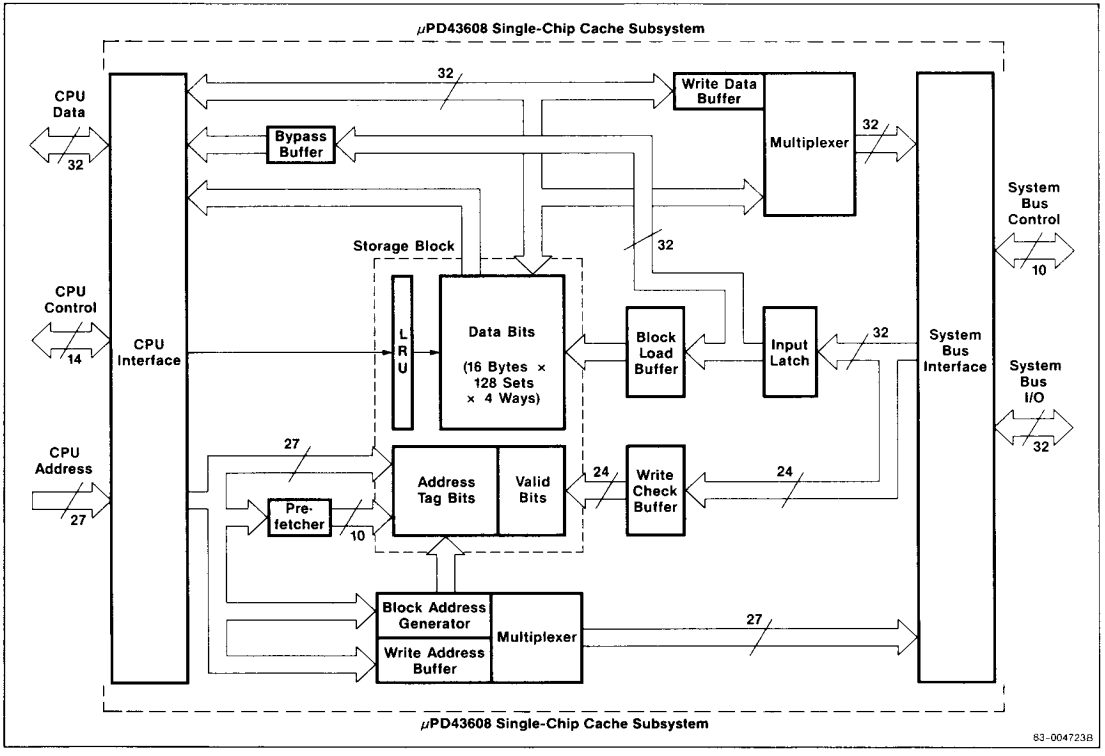
Pin Number	Function
C_6	V_{CC}
C_7	GND
C_8	V_{CC}
C_9	A_4
C_{10}	A_8
C_{11}	A_{11}
C_{12}	A_{14}
C_{13}	A_{16}
C_{14}	A_{20}
D_1	D_{24}
D_2	D_{21}
D_3	D_{18}
D_{12}	A_{15}
D_{13}	A_{19}
D_{14}	A_{21}
E_1	D_{25}
E_2	D_{23}
E_3	V_{CC}
E_{12}	A_{18}
E_{13}	A_{22}
E_{14}	A_{24}
F_1	D_{27}
F_2	D_{26}
F_3	GND
F_{12}	A_{23}
F_{13}	A_{25}
F_{14}	A_{26}
G_1	D_{29}
G_2	D_{28}
G_3	GND
G_{12}	GND
G_{13}	AD_0
G_{14}	A_{27}

Pin Number	Function
H_1	D_{30}
H_2	D_{31}
H_3	GND
H_{12}	GND
H_{13}	AD_2
H_{14}	AD_1
J_1	$\overline{PRDY}$
J_2	$\overline{PAS}$
J_3	$\overline{PCS}$
J_{12}	AD_7
J_{13}	AD_4
J_{14}	AD_3
K_1	$PRD/\overline{PWT}$
K_2	$\overline{CAEN}$
K_3	ST_2
K_{12}	AD_{11}
K_{13}	AD_6
K_{14}	AD_5
L_1	ST_1
L_2	ST_0
L_3	$\overline{PBE_0}$
L_{12}	AD_{14}
L_{13}	AD_{10}
L_{14}	AD_8
M_1	$\overline{PBE_3}$
M_2	$\overline{PBE_1}$
M_3	PCLK
M_4	$\overline{RST}$
M_5	BCLK
M_6	$\overline{MDS}$
M_7	V_{CC}
M_8	GND
M_9	V_{CC}

Pin Number	Function
M_{10}	AD_{21}
M_{11}	AD_{17}
M_{12}	AD_{15}
M_{13}	AD_{13}
M_{14}	AD_9
N_1	$\overline{PBE_2}$
N_2	$\overline{SMC}$
N_3	$\overline{AMC}$
N_4	$\overline{BRQ}$
N_5	$\overline{BACK}$
N_6	$\overline{MBE_2}/\overline{WAIT}$
N_7	AD_{31}
N_8	AD_{29}
N_9	AD_{27}
N_{10}	AD_{24}
N_{11}	AD_{22}
N_{12}	AD_{19}
N_{13}	AD_{16}
N_{14}	AD_{12}
P_1	$\overline{ERR}$
P_2	$\overline{WBSY}$
P_3	$MAS/\overline{MBS}$
P_4	$\overline{MBE_0}/\overline{EOC}$
P_5	$\overline{MBE_1}/\overline{UERR}$
P_6	$\overline{MBE_3}/\overline{CERR}$
P_7	$\overline{MWA}$
P_8	AD_{30}
P_9	AD_{28}
P_{10}	AD_{26}
P_{11}	AD_{25}
P_{12}	AD_{23}
P_{13}	AD_{20}
P_{14}	AD_{18}

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Block Diagram



System Bus Interface

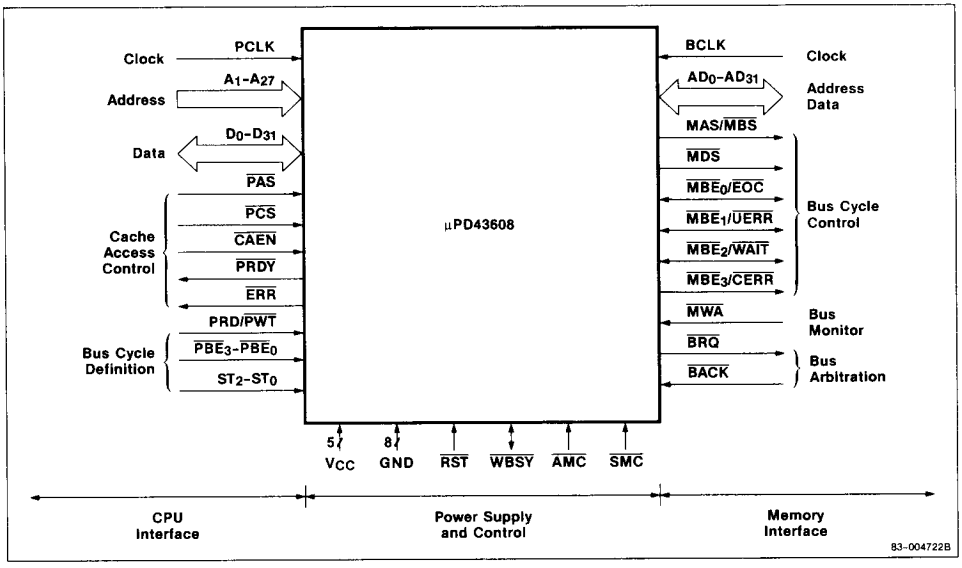
The integrated system bus interface provides an interface to contemporary microprocessor system bus architectures. The interface circuit consists of a 32-bit multiplexed address and data bus, asynchronous bus control signals, a bus lock signal, a wait signal, a correctable error function, two data transfer modes—burst and single, and a system bus clock signal. The size of the cache can be increased by connecting additional μPD43608 devices in parallel. A write buffer busy signal is daisy-chained between the parallel devices and automatically controls data transfers in multichip configurations.

Bus Monitoring

In multiprocessor system applications, maintaining data consistency is a major concern. In such a system architecture, an integrated circuit is required to monitor the system bus for any updates to main storage. When a bus master updates a location in its cache storage and writes that change to main storage, all slave processors must invalidate any stale cache data. The monitoring circuit latches all write addresses on the system bus and invalidates any cache data blocks that are not consistent with main storage.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Functional Pin Diagram



Signal Summary

CPU Interface

Signal Name	Input/ Output	Signal Function
PCLK	I	Processor clock
A_1–A_{27}	I	Address bus
D_0–D_{31}	I/O	Data bus
$\overline{PAS}$	I	Address strobe
$\overline{PCS}$	I	Command strobe
$\overline{CAEN}$	I	Cache output enable
PRD/$\overline{PWT}$	I	Read/write
$\overline{PBE_3}$–$\overline{PBE_0}$	I	Byte enable
ST_2–ST_0	I	Status
$\overline{PRDY}$	O	Ready
$\overline{ERR}$	O	Error

Control

$\overline{RST}$	I	Reset
$\overline{WBSY}$	I/O	Write buffer busy
$\overline{AMC}$	I	Test pin
$\overline{SMC}$	I	Scan path mode

Memory Interface

Signal Name	Input/ Output	Signal Function	
AD_0–AD_{31}	I/O	Address/data bus	
AD_{31} = MEM/$\overline{IO}$	O	Memory/IO	During an address cycle
AD_{30} = MRD/$\overline{MWT}$		Read/write	
AD_{29} = LOCK		Bus lock	
AD_{28} = PRF		Prefetch	
MAS/$\overline{MBS}$	O	Address strobe/bus strobe	
$\overline{MDS}$	O	Data strobe	
$\overline{MBE_0}$/$\overline{EOC}$	O	Byte enable 0/end of cycle	
$\overline{MBE_1}$/$\overline{UERR}$	I/O	Byte enable 1/uncorrectable error	
$\overline{MBE_2}$/$\overline{WAIT}$	I/O	Byte enable 2/wait	
$\overline{MBE_3}$/$\overline{CERR}$	I/O	Byte enable 3/correctable error	
$\overline{MWA}$	I	Main memory write check address	
$\overline{BRQ}$	O	Bus request	
$\overline{BACK}$	I	Bus acknowledge	
BCLK	I	Bus clock	

Powered by ICminer.com Electronic-Library Service CopyRight 2003