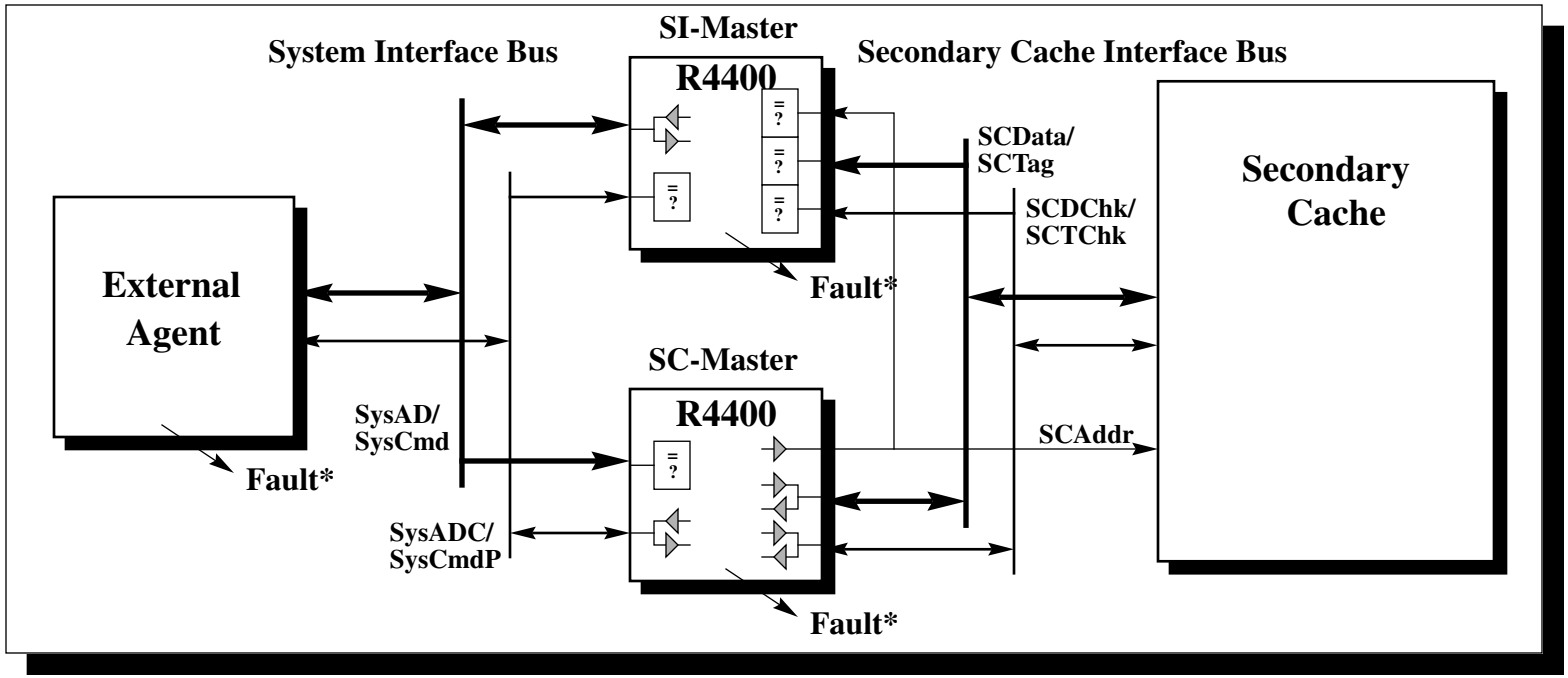
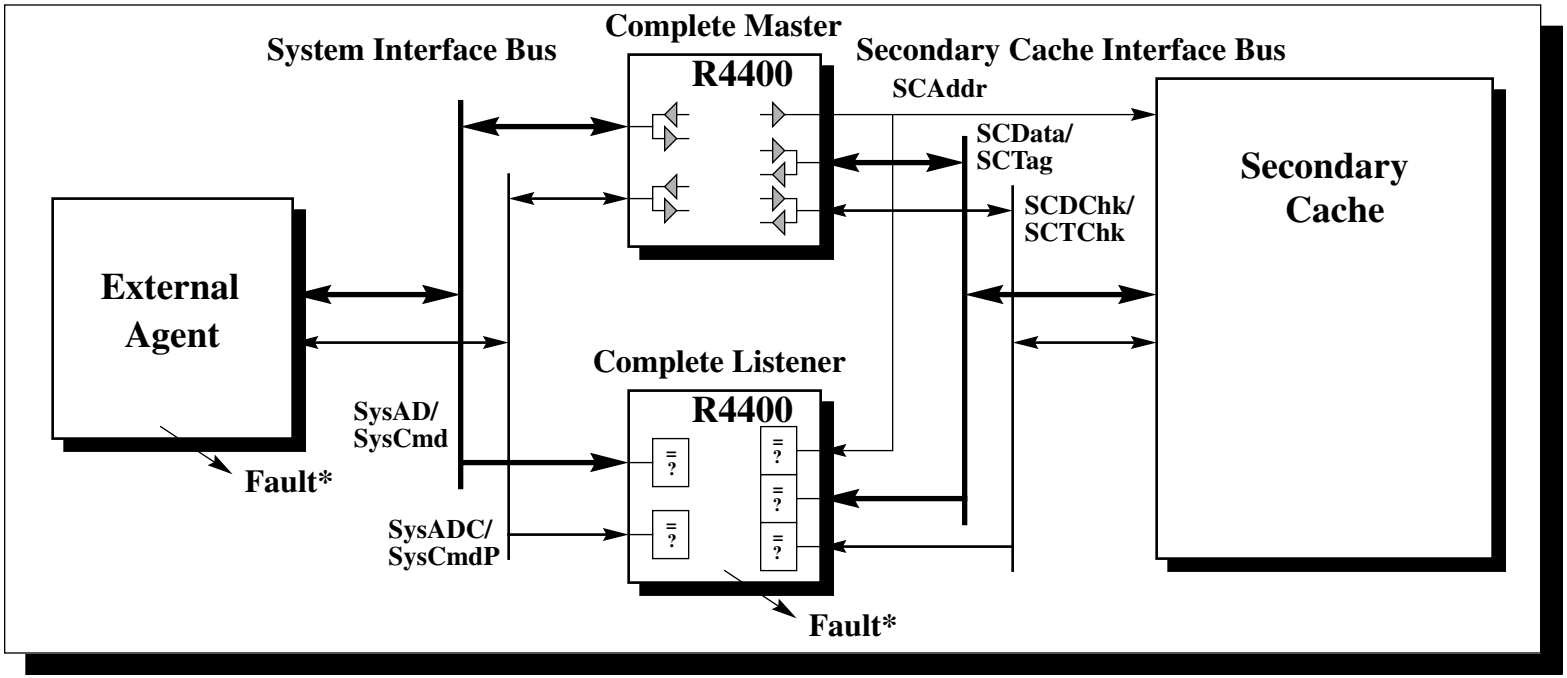
Cross-Coupled Checking Configuration

- In the Cross-Coupled Checking configuration, one of the R4400 processors, the SI-Master, drives the system interface bus, while the other R4400 processor, the SC-Master, drives the secondary cache bus
- The SI-Master has disabled output drivers for the secondary cache interface bus while the SC-Master has disabled output drivers for the system interface bus
- The two processors operate identically, both receiving the same input
- On all output cycles, both processors monitor the buses and indicate a miscomparison by asserting the **Fault*** signal

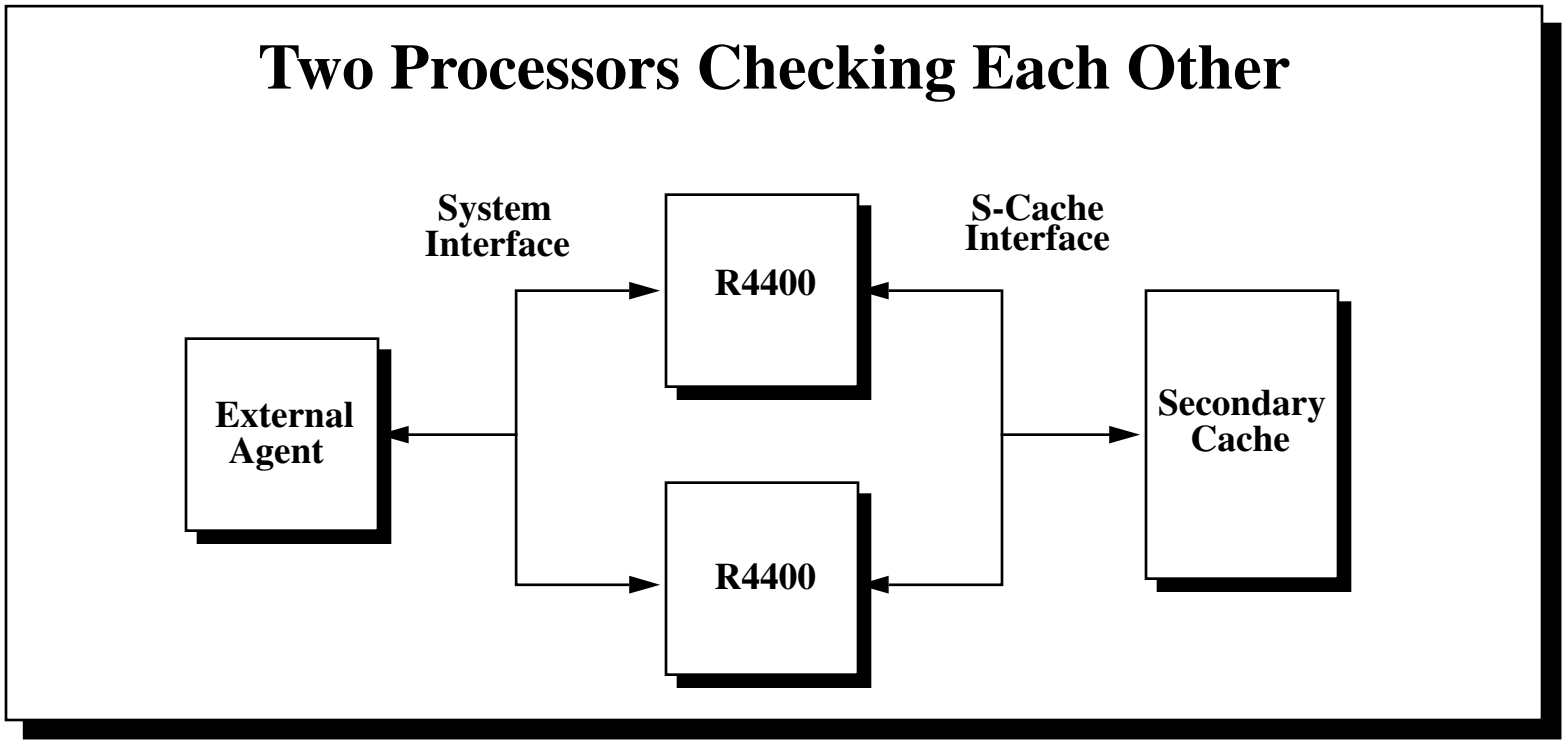


Master-Listener Configuration

- The Master-Listener lock step configuration pairs two R4400 RISC-processors, with one as a Complete Master and the other as a Complete Listener
- The Complete Listener has disabled output drivers
- The two processors operate identically, both receiving the same inputs
- On all output cycles, the Complete Listener compares data on the output and the I/O buses with expected data, and asserts the **Fault*** signal in the event of a miscomparison



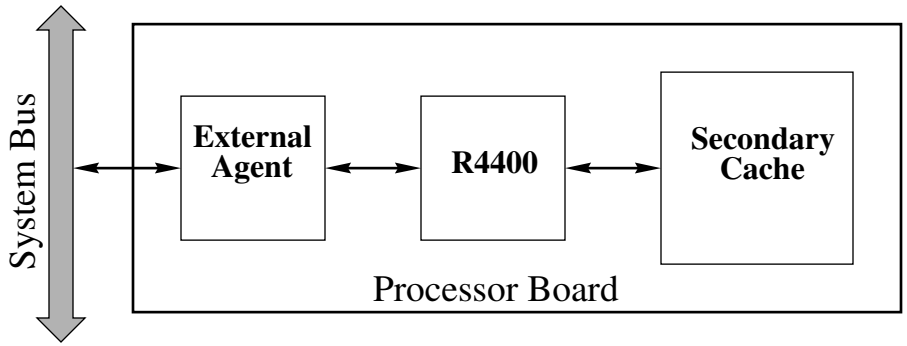
Master-checker



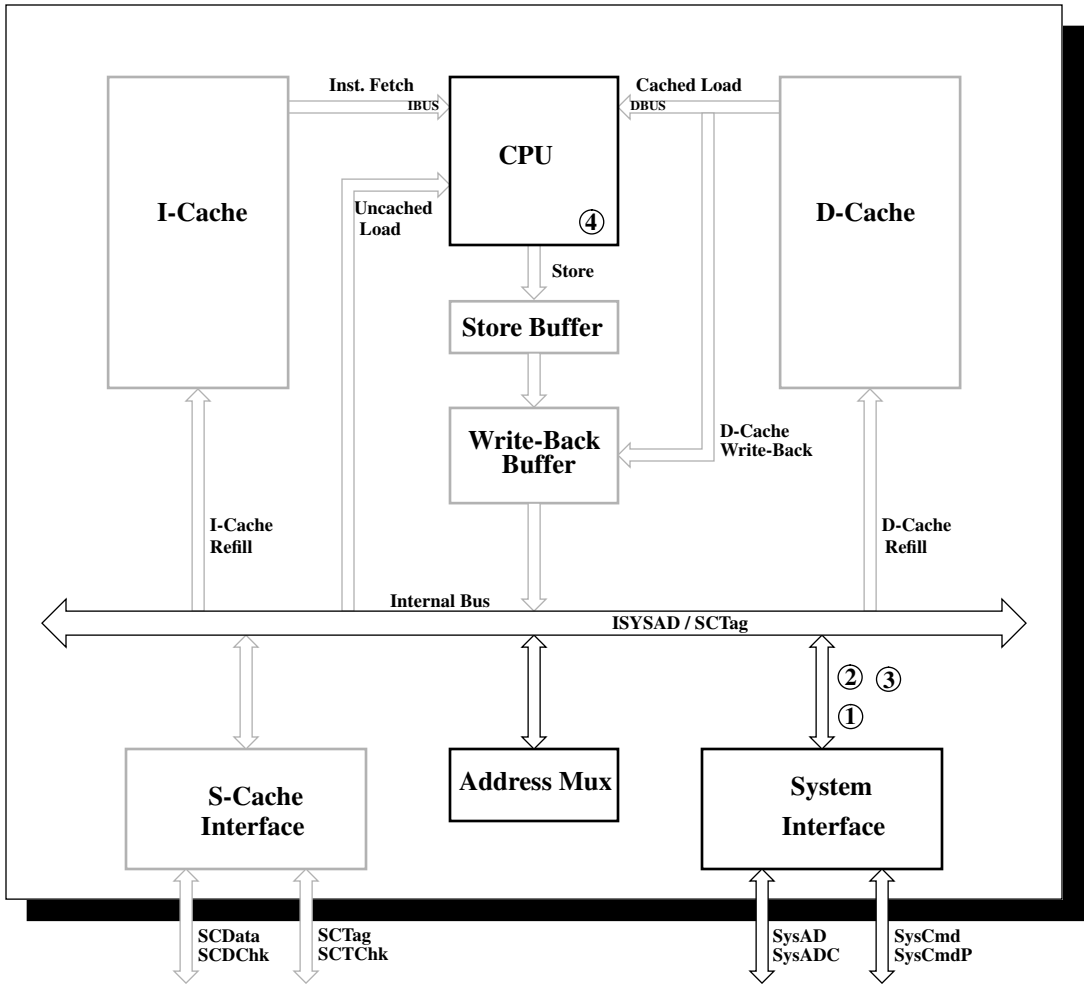
- Two processors are connected in parallel (one is the bus driver while the other is the bus monitor)
- Processor synchronization
- Two possible configurations: Master-Listener and Cross-Coupled

Cache Error Recovery

- **Single-bit error corrector**
  - Secondary cache tag ECC has a single-bit error corrector which is used to correct a tag error during external intervention and snoop requests. The corrected tag is written back to the secondary tag array only when the tag needs to be modified.
  - No single-bit error corrector for secondary cache data
  - External single-bit error corrector
- **Cache Error Exception and status**
  - Taken when the processor detects any cache parity or ECC error
  - Error status is saved in CP0 CacheErr register
- **Cache Error Exception handler**
  - Uses the information saved in the CacheErr register to perform single-bit error corrections in the secondary cache Data or Tag portion



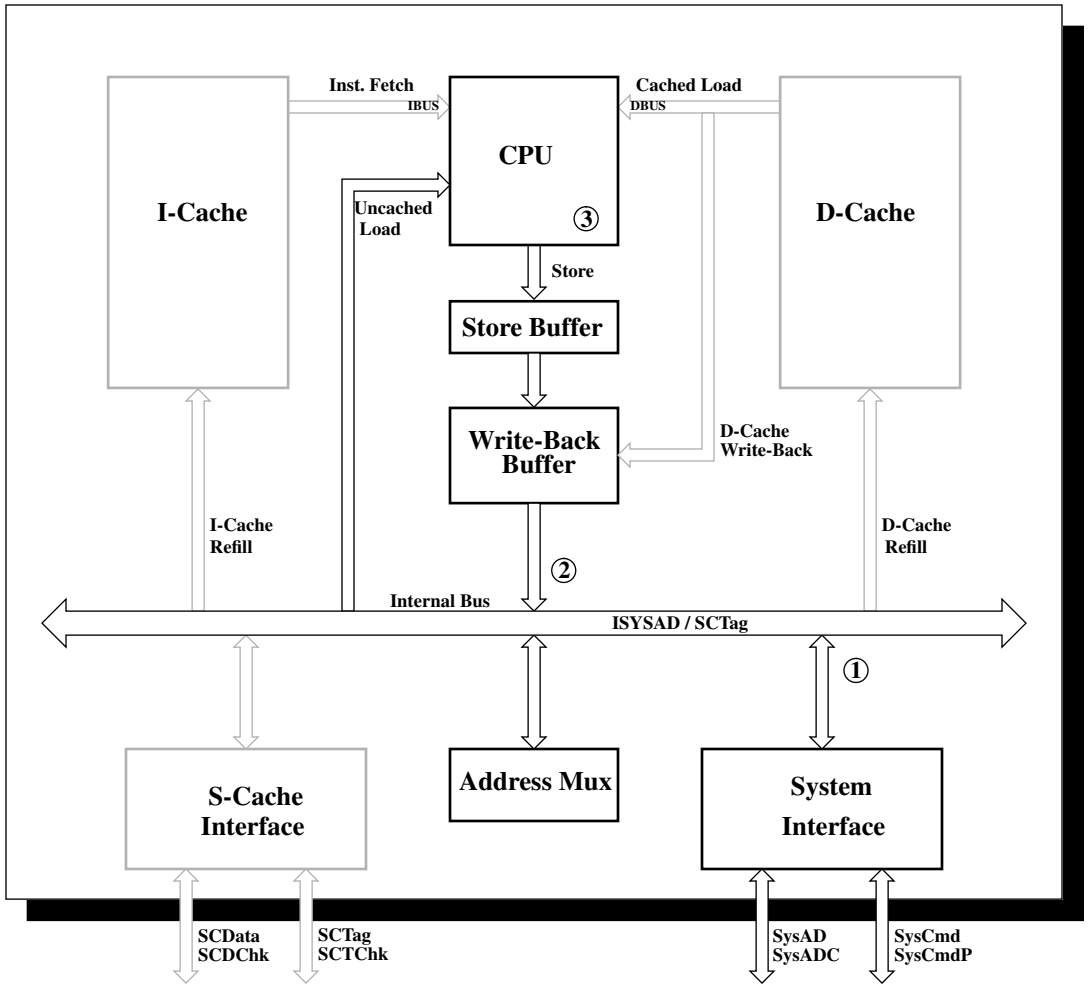
External Read/Write Requests



1. The address and command portions of the in-coming request are not checked
2. Parity or ECC checking is performed on in-coming data from the system interface[1]
3. Data check bits are generated for out going data before being transferred to the system interface
4. The processor will take a Bus Error Exception when a parity or ECC error is detected on in-coming data

1. Byte parity or ECC checking is performed based on the selected data protection scheme for the system interface

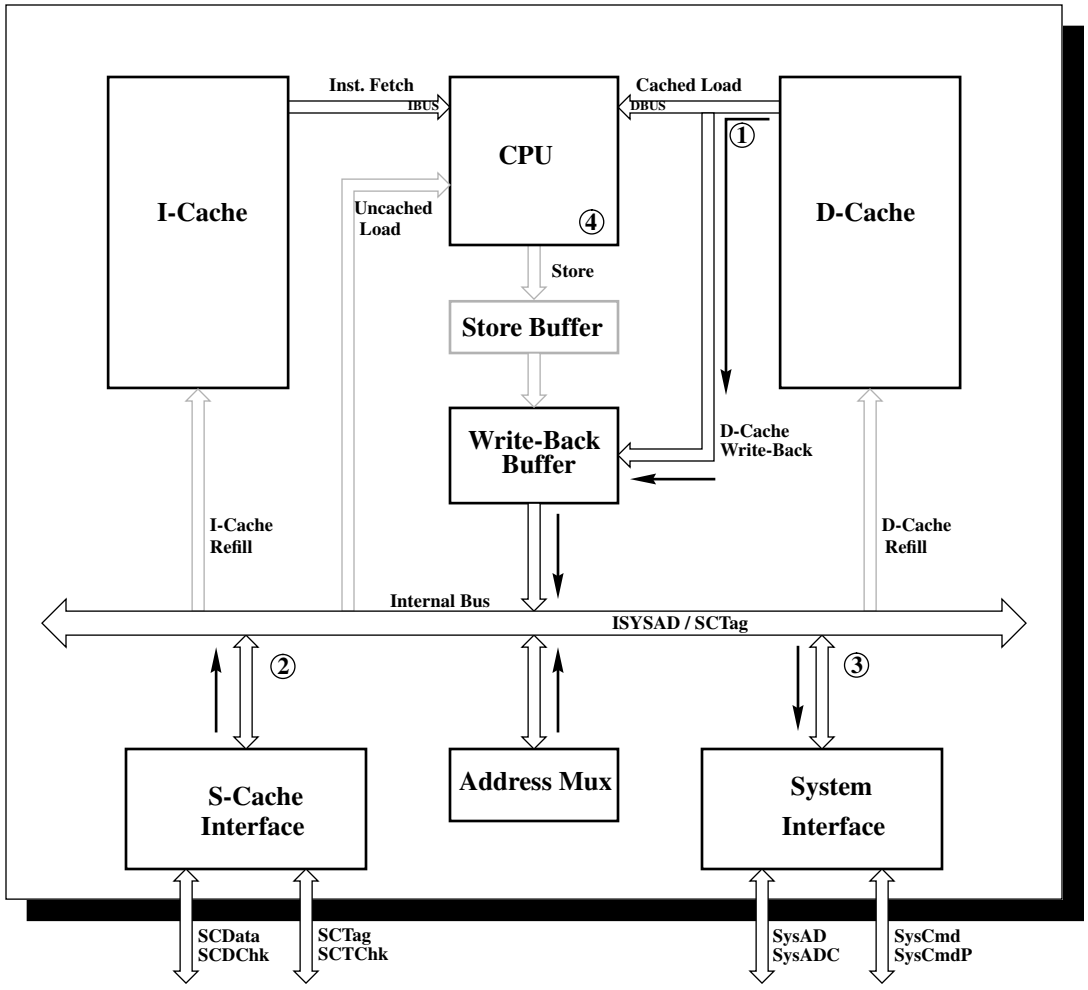
Processor Uncached Load/Store Operations



1. Parity or ECC checking is performed on uncached load data coming from the system interface[1]
2. For store data, the appropriate type of check bits will be generated before transferred to the system interface bus
3. The processor will take a Bus Error Exception when a parity or ECC error is detected on in-coming data

1. Byte parity or ECC checking is performed based on the selected data protection scheme for the system interface

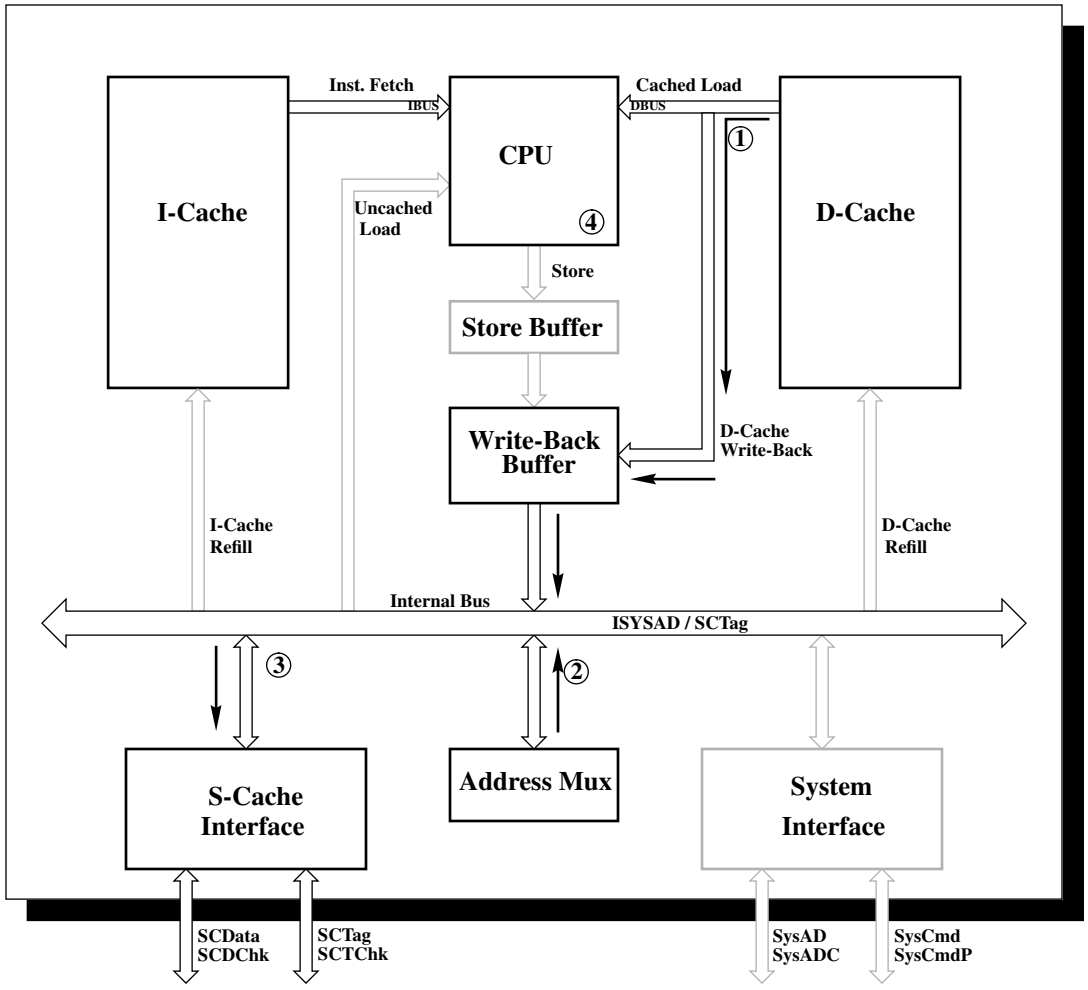
Primary/Secondary Cache Write-back to System Interface



1. Parity checking is performed on Data and Tag portions of the selected primary cache line
2. ECC checking is performed on the Data and Tag portions of the selected secondary cache line
3. Check bits are generated for the data transfer to the system interface[1]
4. The processor will take a Cache Error Exception when a parity error is detected in the primary cache Data or Tag portion or ECC error is detected in the secondary cache Data or Tag portion

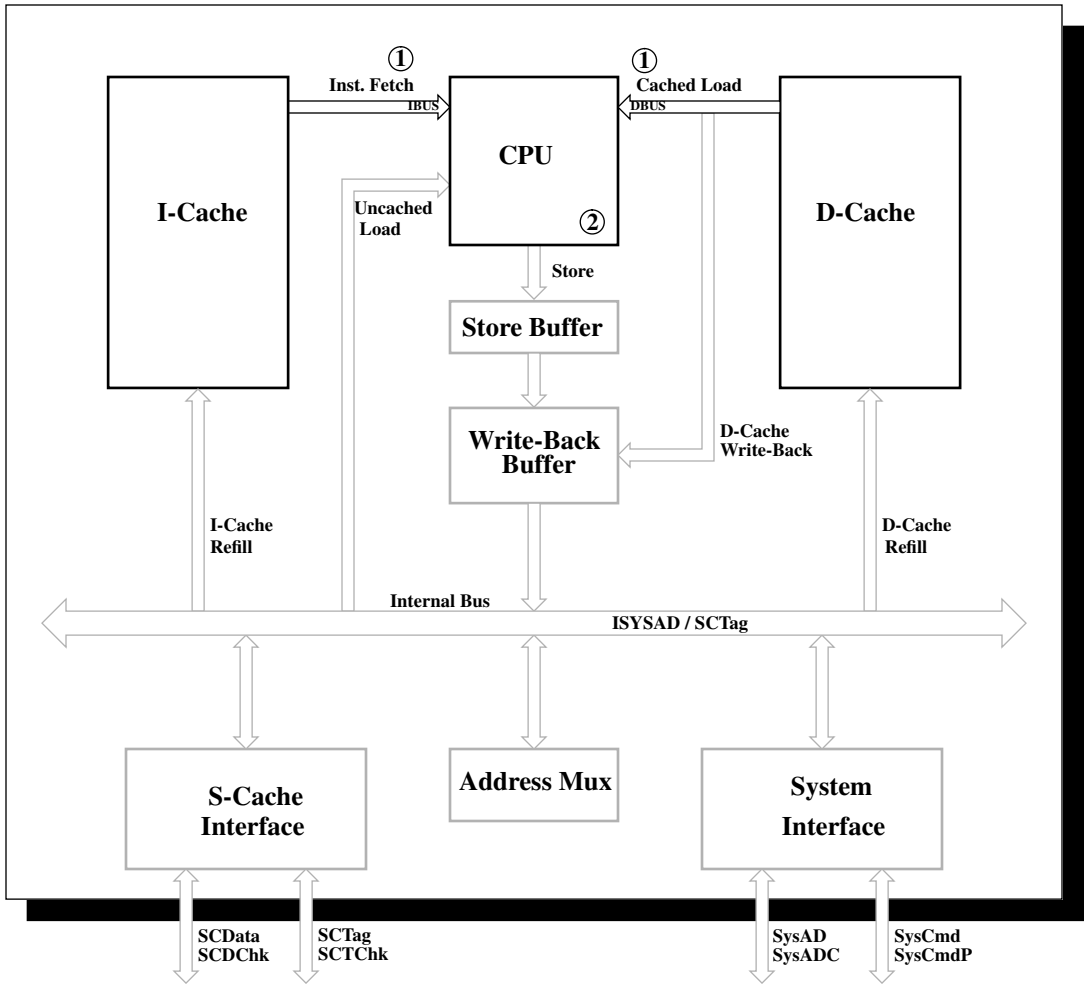
1. The appropriate type of check bits will be generated if they do not match the selected data protection scheme for the system interface. Otherwise, they will be transferred to the system interface unchanged.

Primary Cache Write-back to Secondary Cache



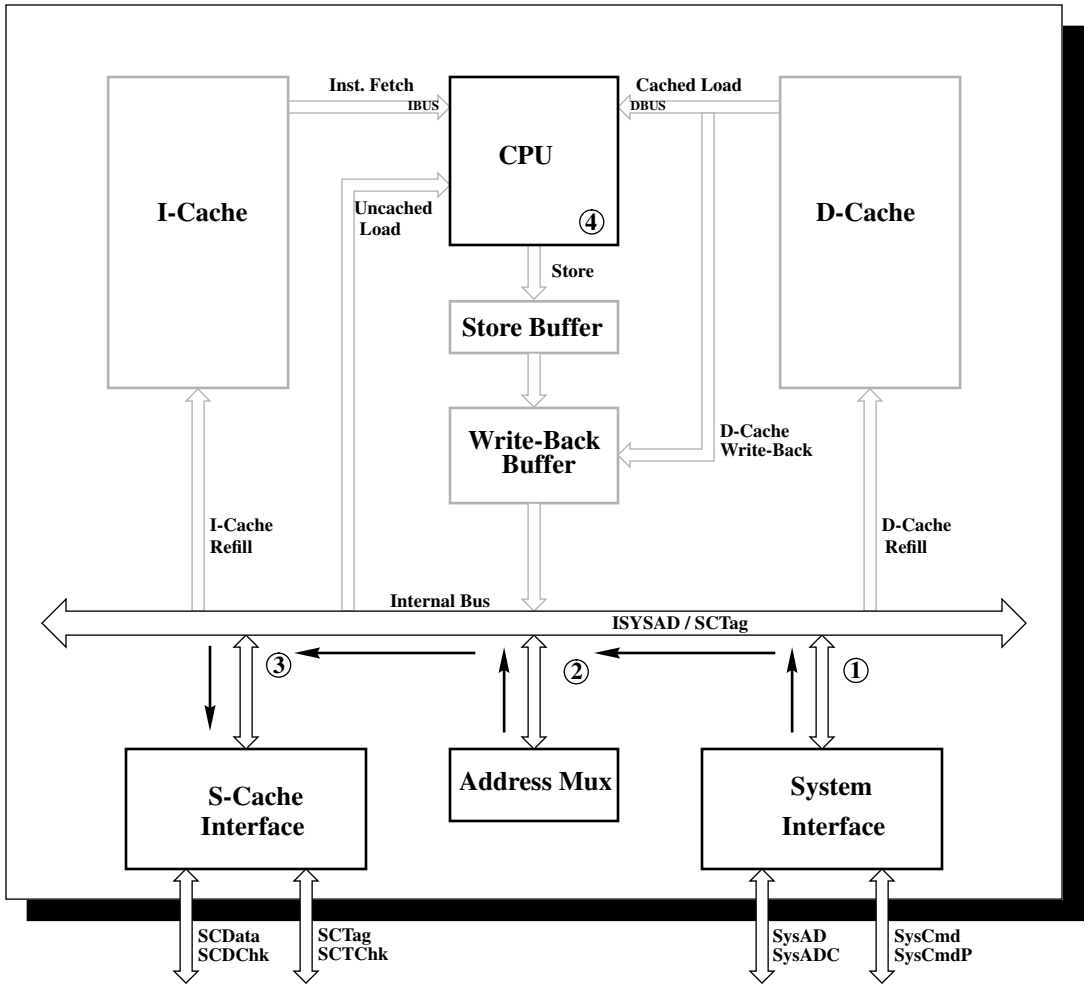
1. Parity checking is performed on Data and Tag portions of the selected primary cache line
2. Secondary cache tag's ECC check bits are generated for the line being updated
3. Data ECC check bits are generated for the line being updated in the secondary cache
4. The processor will take a Cache Error Exception when a parity error is detected on the primary cache Data or Tag portion

Primary Instruction/Data cache Access



1. Parity checking is performed on Data and Tag portions of the selected primary cache line
2. The processor will take a Cache Error Exception when a parity error is detected on the primary cache Data or Tag portion

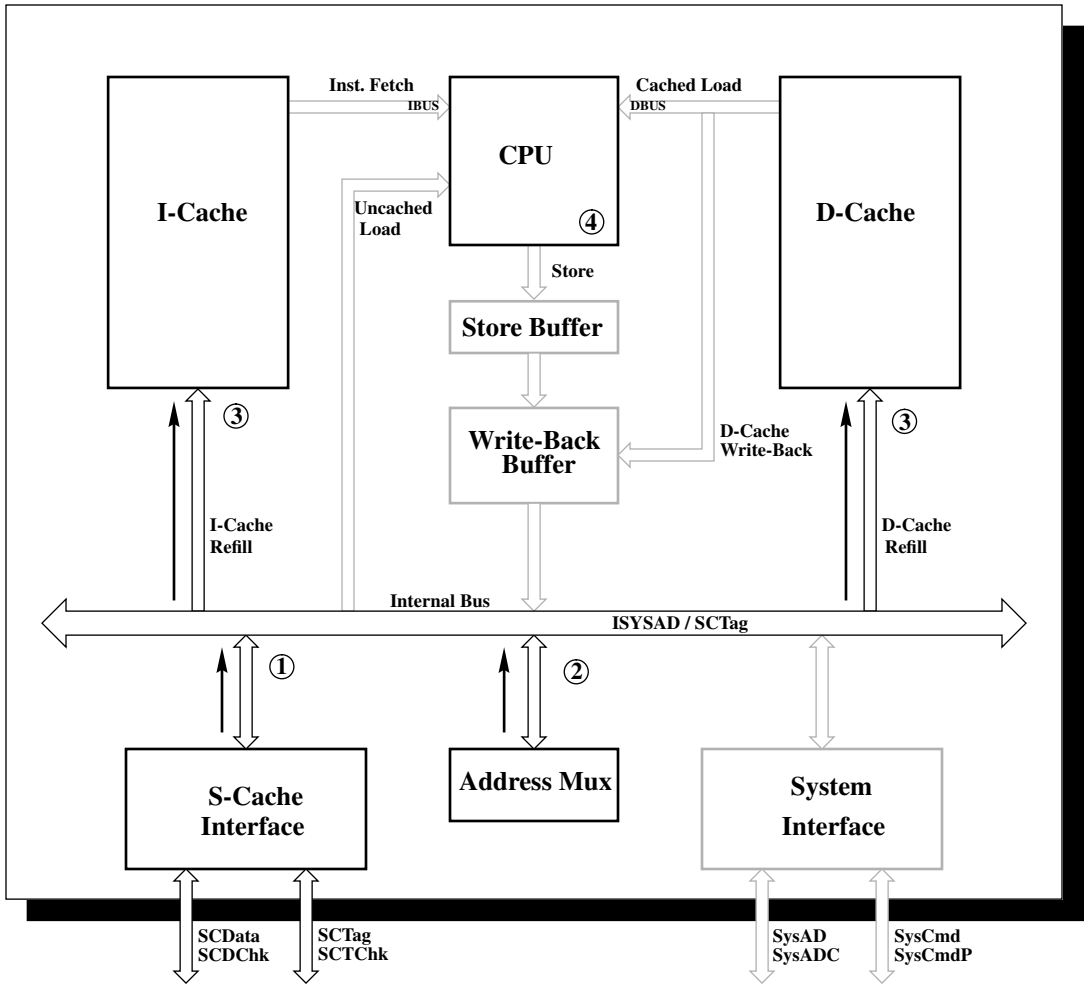
Secondary Cache Refill



1. Performs parity or ECC checking on data coming from the system interface[1]
2. Secondary cache Tag and Tag's ECC check bits are generated for the line being refilled
3. Data ECC check bits are generated for the line being updated in the secondary cache[2]
4. The processor will take a Cache Error Exception when an error is detected on the in-coming data from the system interface bus

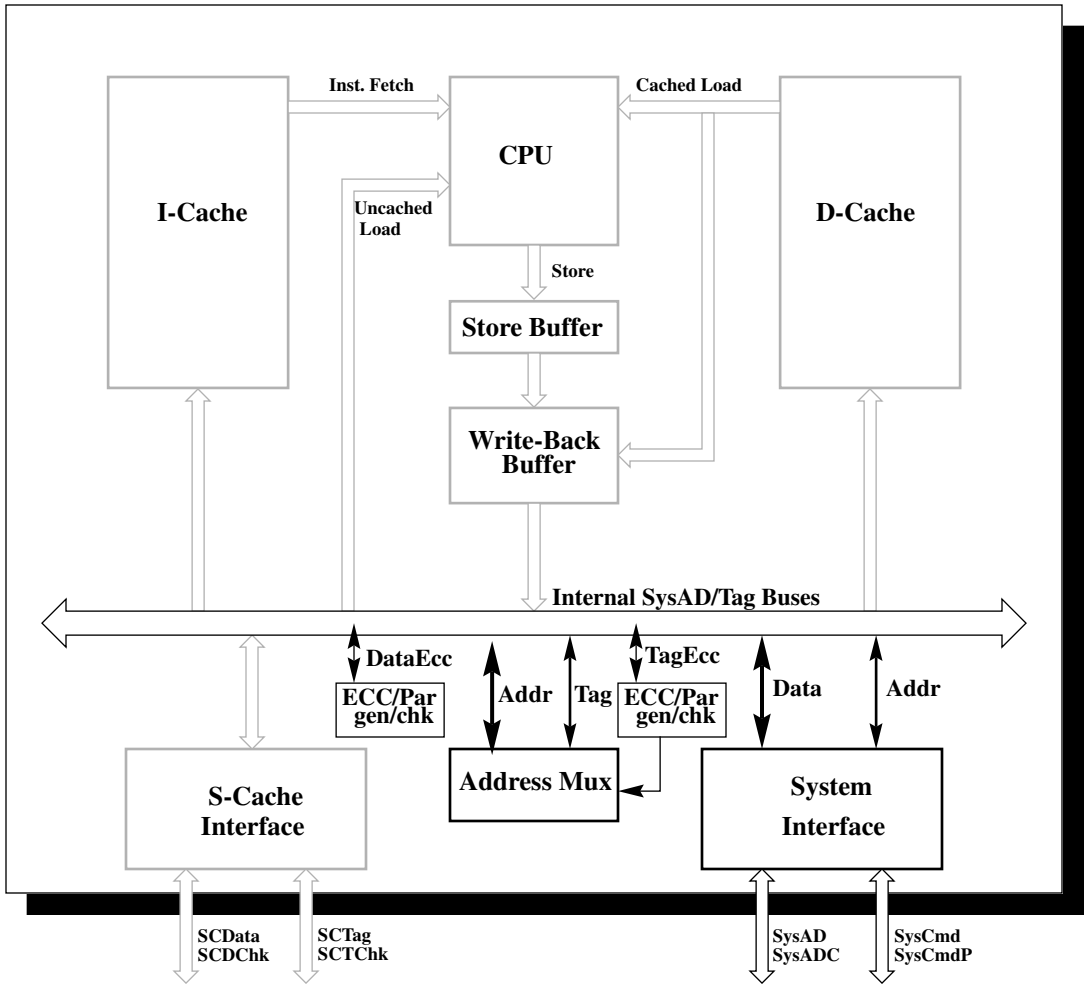
1. Byte parity or ECC checking is performed based on the selected data protection scheme for the system interface
2. If the system interface bus is programmed to use ECC protection scheme then the in-coming check bits will be written to the secondary cache unchanged

Primary Instruction/Data Cache Refill



1. ECC checking is performed on the secondary cache Data and Tag
2. Primary cache Tag and Tag's parity are generated for the line being refilled
3. Data byte parity bits are generated for the line being refilled in the primary cache
4. The processor will take a Cache Error Exception when an error is detected in the secondary cache Data or Tag portion

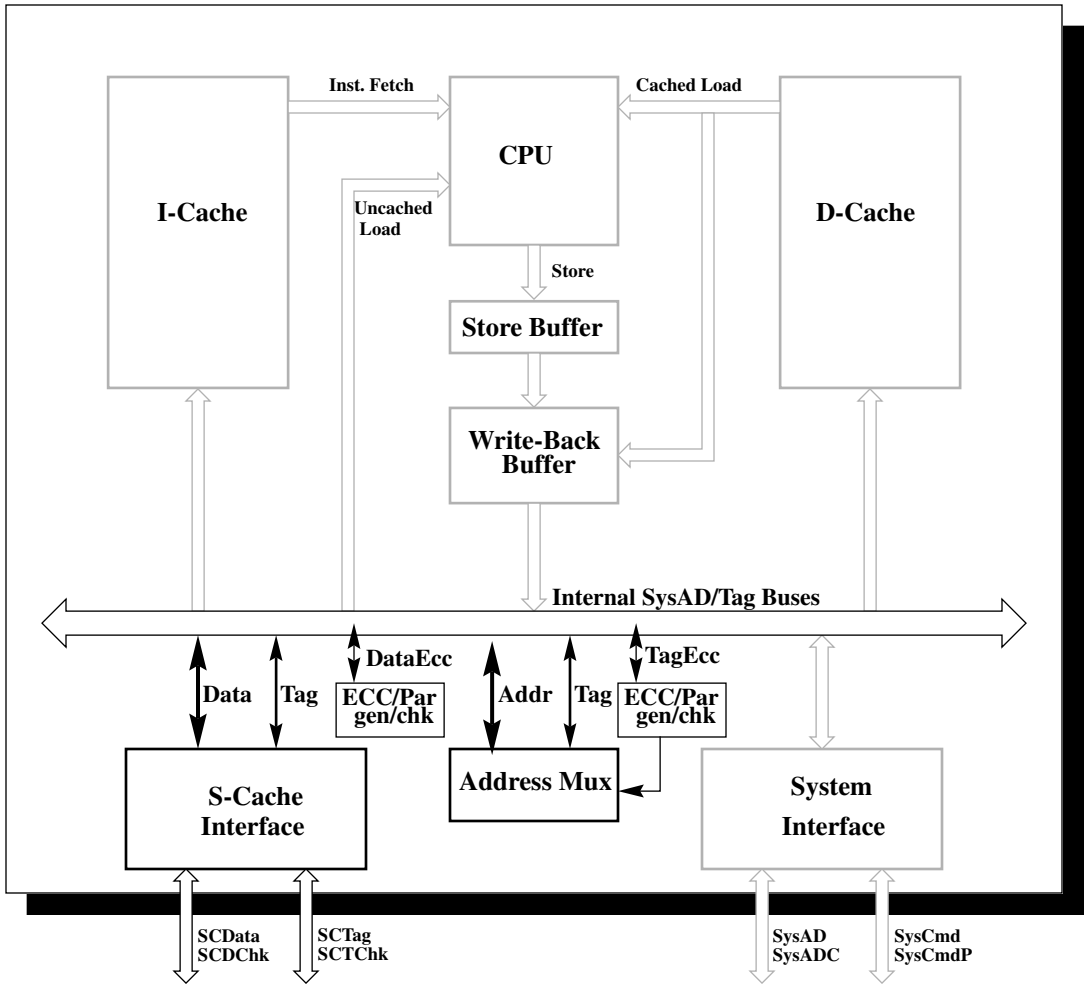
Error Checking on System Interface



- **Parity or ECC protected on SysAD bus[1]**
- **Parity protected on SysCmd bus**
- **For out-going transactions:**
  - Parity or ECC check bits are generated for the address and parity is generated for the command
  - Data's parity or ECC check bits come from primary cache, secondary cache or generated for the processor store data[2]
- **For in-coming transactions:**
  - Address and command portions are not checked
- **Parity or ECC checking is performed for in-coming data[3]**

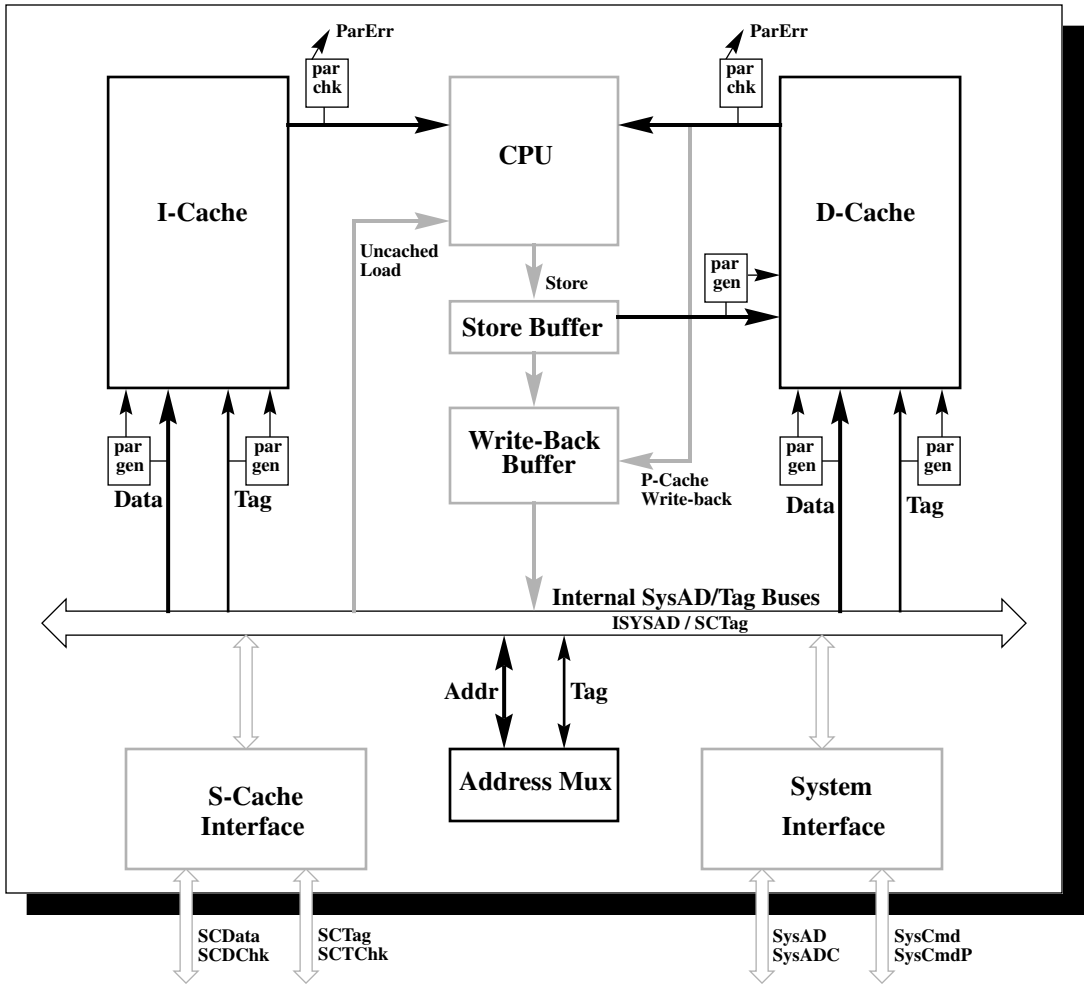
1. Programmable via boot-mode
2. The appropriate type of check bits will be generated if they do not match with the selected data protection scheme for the system interface
3. In-coming data will not be checked for external invalidate or update request

Error Checking in Secondary Cache



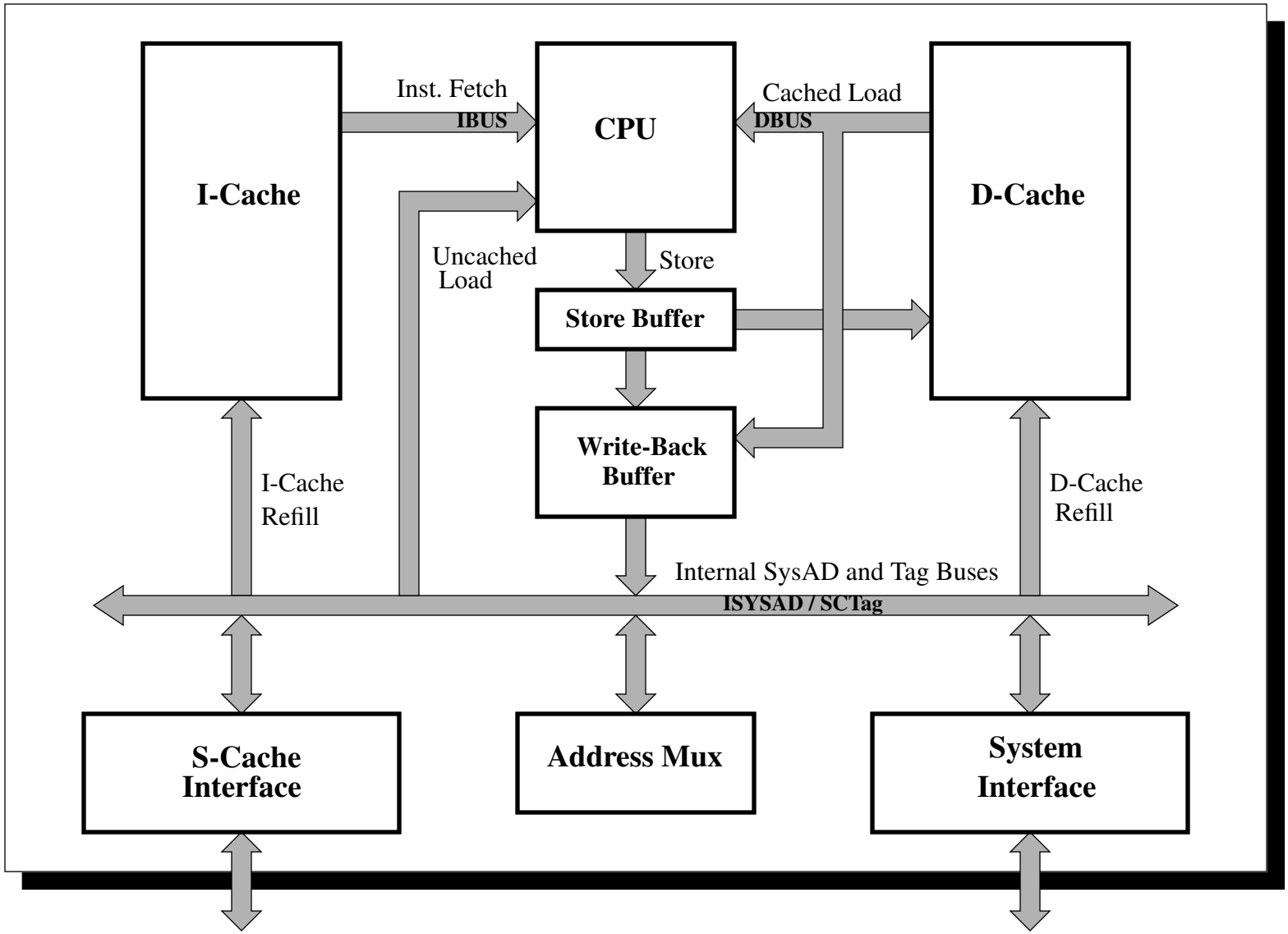
- ECC protected on Data Array
- ECC protected on Tag Array
- Secondary cache address is parity protected
- ECC check bits for the Data and Tag portions are generated on secondary cache update
- ECC checking is performed on every secondary cache access (primary instruction/data cache refill or secondary cache write-back)

Error Checking in Primary Instruction/Data Caches



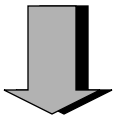
- Byte parity protected on Data Array
- Word parity protected on Tag Array
- Parity is generated on primary cache write
- Parity checking is performed on every primary cache access (instruction fetch, data fetch, or cache write-back)

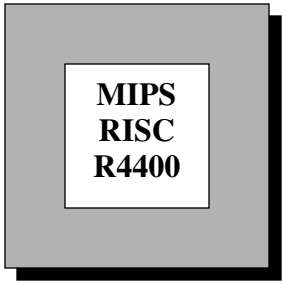
Internal Cache Memory and Datapath



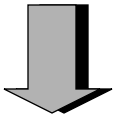
R4400 Data Integrity

Detection of data corrupted due to soft errors caused by system noise, power surges, and alpha particles





- Large on-chip static RAM
- Reliability
- Fault tolerant system design



- Parity checking on the primary instruction and data caches
- ECC checking on the secondary cache
- Parity or ECC checking on the system interface bus
- Master-checker

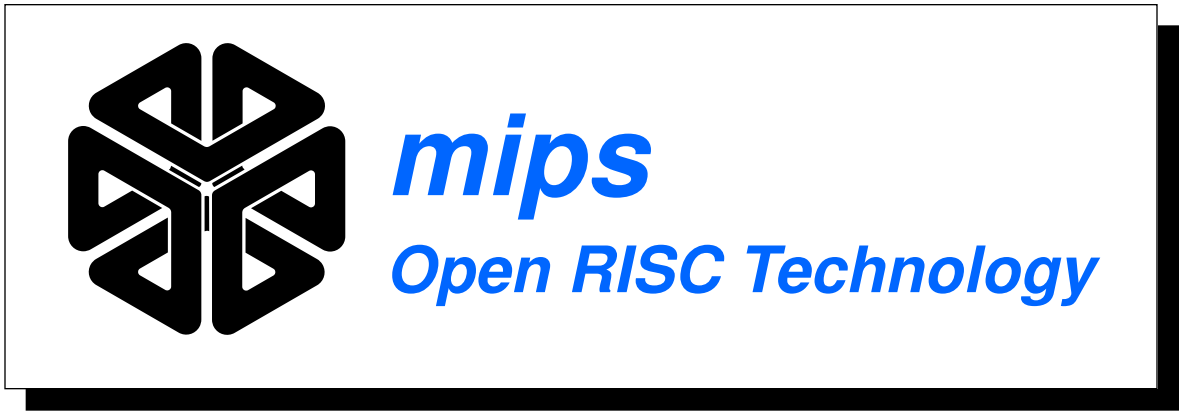
R4400 Data Integrity

Agenda:

- Internal datapath
- Data flow
- Error checking
- Error recovery
- Master-checker

R4400 Data Integrity

Presented by: Michael Ngo



MIPS Technologies, Inc.
A Technology Company of Silicon Graphics, Inc.
2011 N. Shoreline Blvd.
P.O. Box 7311
Mountain View, CA 94039-7311
Phone: (415) 960-1980