CACHE FEATURES

ARCHITECTURE
- SEPARATE I & D
- PHYSICAL CACHE
- DIRECT MAPPED
- WRITE THROUGH

ON-CHIP CACHE CONTROL
- BLOCK REFILL (1, 4, 8, 16, 32 WORDS)
- INSTRUCTION STREAMING
- DIRECT DRIVE OF SRAMS

IMPLEMENTATION
- USES STANDARD SRAMS
- WIDE RANGE OF SIZES & SPEEDS
- WRITE BUFFERS

PERFORMANCE ON M20000 SYSTEM:
- MISS RATE .005 FOR BLOCK of 16
  (FOR MIPS BENCHMARK)
CACHE ISSUES

PHYSICAL OR VIRTUAL

DIRECT MAPPED OR SET ASSOCIATIVE

STANDARD OR SPECIAL SRAMS

BLOCK REFILL SIZES AND STREAMING
ADVANTAGE OF PHYSICAL CACHE

- No synonyms
- No page coloring
- No reverse translations
- Reduces cache flushing
- No cycle time penalty in pipeline
- 1/2 cycle on-chip translation
- Full cycle for cache access
- Well-suited for multiprocessor design
MINIMIZING CACHE EFFECTS

- PARTIAL WORD STORE

- BLOCK REFILL

- INSTRUCTION STREAMING
PARTIAL WORD STORE

- KEEPS CACHE ENTRY VALID

- PARTIAL WORD STORES USE READ

-MODIFY-WRITES- ELIMINATES SEPARATE WRITE CONTROLS
MIPS BLOCK REFILL

• AMORTIZE COST OF LATENCY OVER MULTIPLE WORDS

• CPU HARDWARE CONTROLS THE TRANSFER

• CHOICE OF 4, 8, 16, or 32 WORD BLOCK SIZES

• SEPARATE INSTRUCTION AND DATA BLOCK SIZES

• DESIRABLE TO TUNE BLOCK SIZE TO SYSTEM DESIGN, INCREASE BLOCK SIZE AS LATENCY TO MEMORY INCREASES

• BLOCK REFILL HAS LARGE IMPACT ON HIT RATE, SECOND ONLY TO TOTAL CACHE SIZE
MIPS INSTRUCTION STREAMING

MISS    |    LATENCY    |    BLOCK REFILL    |    CONTINUE

MISS    |    LATENCY    |    BLOCK REFILL    |    CONTINUE
MIPS INSTRUCTION STREAMING

- BEGIN EXECUTION WHEN INSTRUCTION ARRIVES, ISSUE INSTRUCTION IN SAME CYCLE AS CACHE WRITE

- 70% OF INSTRUCTION MISSES START AT BLOCK BOUNDARY, GREATEST BENEFIT TO STREAMING INSTRUCTIONS

- MAXIMUM EXECUTION SPEED MATCHES MAX MAIN MEMORY BANDWIDTH, 100MB/SEC

- 5% NET GAIN - vs SYSTEM WITHOUT INSTRUCTION STREAMING
LR3000 CACHE DESIGN
LR3000 CACHE DESIGN
LR3000 CACHE DESIGN

• Cache Introduction

• LR3000 Cache Interface

• Determining the Required Clock Delays

• Small Systems
WHAT IS A CACHE?

- Fast memory between the processor and the main memory
CONSIDERATIONS FOR A CACHED SYSTEM

• Cacheable accesses with hits improve system performance
• Misses result in penalties
• Mapping strategies - Direct, Set associative, Fully associative
• Replacement strategies - LRU, Random, LFU, etc.
• Needs initialization
MAPPING STRATEGIES

Direct

Fully Associative

Cache       Memory

Cache       Memory
CACHE INTERFACE

LR3000

AddrLo

Addr Latch

OE

FAST SRAM
(20ns for 25 MHz)

Clk
Rd
Wr

Tag

Data

10-16

28

32
## SUPPORTED CACHE SIZES

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Words</th>
<th>AddrSize</th>
<th>AdrLo Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 K</td>
<td>1024</td>
<td>10</td>
<td>11:2</td>
</tr>
<tr>
<td>8 K</td>
<td>2048</td>
<td>11</td>
<td>12:2</td>
</tr>
<tr>
<td>16 K</td>
<td>4096</td>
<td>12</td>
<td>13:2</td>
</tr>
<tr>
<td>32 K</td>
<td>8192</td>
<td>13</td>
<td>14:2</td>
</tr>
<tr>
<td>64 K</td>
<td>16384</td>
<td>14</td>
<td>15:2</td>
</tr>
<tr>
<td>128 K</td>
<td>32768</td>
<td>15</td>
<td>16:2*</td>
</tr>
<tr>
<td>256 K</td>
<td>65536</td>
<td>16</td>
<td>17:2*</td>
</tr>
</tbody>
</table>

* Requires Extended Cache (ExCache) Mode and Uniprocessor Configuration

64KBytes is Assumed for this Class
CACHE LOOK-UP (SIMPLIFIED)

- A Hit Also Requires Both Parity Fields to be Valid and the Valid Bit to be Set
CACHE LINE FORMAT

Cached Main Memory Data

Data Parity

Location of Main Memory Data

Valid Cache Line

Tag and Valid Bit Parities

0 32 31 32 4 1 20 3 59

8/14/91 cache - 19
64KBYTE CACHE IMPLEMENTATIONS

16Kx4 (15):

8Kx20 (6):
REDUNDANT TAG BITS

- Tag Identifies Where the Cache Word Came From in Main Memory
- Smaller Main Memory Requires Less Tag Bits
- Larger Cache Requires Less Tag Bits

Tag

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 |

- Only Used for Large Main Memory
- Required Tag
- Only Used for Smaller Caches
SMALL SYSTEMS

EXAMPLE: 1MByte of Physical Memory, 64KByte Caches

- Each Cache Line can Come From One Of 16 Places (Tag Size = 4)
- Bootstrap Vectors Start at 1FC0_0000

1FCF_FFFF

1MB MAIN MEMORY

1FC0_0000

64KB ICACHE

64KB DCACHE

Remaining PFN Bits Must Drive a Constant on CPU Reads

DATA PV PFN 44
REDUNDANT TAG BITS CALCULATION

EXAMPLE: 1MB Main Memory & 64KB Caches; Tag = 16..19
TAG ORGANIZATION (SMALL SYSTEM)

EXAMPLE: 1MByte of Physical Memory, 64KByte Caches

- Use FCT241A to Drive Tag Bus
- Save 4 RAM Packages (16K x 4) per Cache
- No RAM Package Savings for 8K x 20
SUPPLYING THE REDUNDANT TAG BITS

- Processor Should Store Start Address of Main Memory in UTAG Register
SMALL SYSTEMS

EXAMPLE: 1MByte of Physical Memory, 64KByte Caches
SMALL SYSTEMS

EXAMPLE: 1MByte of Physical Memory, 64KByte Caches and No Parity Requirement

No Parity

LR3000 → LR3000A

- LR3000A Allows Elimination of an Extra 16K x 4 Package or 2 8K x 20 Packages
**REDUNDANT TAG ENTRIES**

- Tag is Same for All Entries of a Block
- Tag Memory Depth can be Reduced

<table>
<thead>
<tr>
<th>BLOCK ADDRESS</th>
<th>DATA + 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA + 1</td>
</tr>
<tr>
<td></td>
<td>DATA + 2</td>
</tr>
<tr>
<td></td>
<td>DATA + 3</td>
</tr>
</tbody>
</table>
REDUNDANT TAG ENTRIES

EXAMPLE: 1MByte Physical Memory, 64KByte Caches, 4 Word Block Size

LR3000

AdrLo[15:2]

[3:2]

[15:4]

4Kx4
TagP + V

4Kx4
Tag

16Kx4
DataP

16Kx32
Data

Data +
Tag +
TagP +
DataP +
V
REDUNDANT TAG ENTRIES

Useful Only for ICache !!!
SMALL SYSTEMS

EXAMPLE: 128KByte of Physical Memory, 64KByte Caches

- Increased Performance if Application Can Fit in Cache
• Cache Design is Choosing Correct Clock Delays to Permit Reliable Data Transfer Among the Three Major System Components
THE FOUR PHASE CLOCK

CLK 2xSys
CLK 2xSmp
CLK 2xRd
CLK 2xPhi

OSC
DELAY LINE

CLK 2x
Sys
Smp
Rd
Phi

t Cyc

t Sys

t Rd

t Smp
TWO DESIGN APPROACHES

1. Use "Riordan" Formulas to Guarantee Correct Timing for Each Parameter.

2. Make a Few Assumptions and use Simple 3 Step Process to Set Clock Delay Values
RIORDAN CLOCK DELAY CALCULATIONS

Diagram showing logic for calculating clock delay with MAX and MIN functions.
# CALCULATING \( t_{Smp} \)

<table>
<thead>
<tr>
<th>FORMULA</th>
<th>DESCRIPTION</th>
<th>FOR 25MHz ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>Internal Sample to Phase Delay</td>
<td>5</td>
</tr>
<tr>
<td>r2</td>
<td>Address Capture by IClk &amp; DClk</td>
<td>1.8</td>
</tr>
<tr>
<td>r3</td>
<td>Address Access to Sample</td>
<td>6.8</td>
</tr>
<tr>
<td>r4</td>
<td>Address Setup to End of Write</td>
<td>19.8</td>
</tr>
<tr>
<td>r5</td>
<td>Processor Data Setup to End of Write</td>
<td>7</td>
</tr>
</tbody>
</table>
Address Capture by IClk and DClk - This constraint guarantees that the transparent latches on the AdrLo bus close before the processor changes the address.

\[
\text{t Smp} = t_{373\text{Hld}} + t_{\text{ClkDeMin}} - t_{\text{AdrLoDeMin}} = 1.8 + 0.5 - 0.5 = 1.8
\]

- SSI Logic and Cache Ram Propagation Delay Derated by 1ns
Address Capture by IClk and DCIkl - This constraint guarantees that the transparent latches on the AdrLo bus close before the processor changes the address.
## OUTPUT DERATING

Derate 1ns/25pF beyond 25pF (Standard Load)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>t374OeDe</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>tRamAaDe</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>tRamDoeDe</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>tDValDe</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>tIRdDe</td>
<td>0.5</td>
<td></td>
</tr>
</tbody>
</table>

2-3.5pF/Inch

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>tClkDe</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>tAdrLoDe</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>tDRdDe</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>tRdDe</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>tSysDe</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>tWrDe</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>tXEnDe</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
CALCULATING \( t \text{ Smp} \)

- Round Up the Min to the Next Even Number

\[
\begin{align*}
r_1 &= 5 \\
r_2 &= 1.8 \\
r_3 &= 6.8 \\
r_4 &= 19.8 \\
r_5 &= 7
\end{align*}
\]

\[
\begin{align*}
\text{MAX} & \quad 5 \\
\text{MIN} & \\
\text{SELECT} & \\
t \text{ Smp} &= 6
\end{align*}
\]
<table>
<thead>
<tr>
<th>FORMULA</th>
<th>DESCRIPTION</th>
<th>FOR 2.5MHz ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>r20</td>
<td>Read-Write, ICache-Data Bus Contention</td>
<td>10.5</td>
</tr>
<tr>
<td>r21</td>
<td>Read-Write, Read Buffer-Data Bus Contention</td>
<td>7</td>
</tr>
</tbody>
</table>
CALCULATING $t_{Sys}$

- Round Up the Min to the Next Even Number
## CALCULATING t Rd Min

<table>
<thead>
<tr>
<th>FORMULA</th>
<th>DESCRIPTION</th>
<th>FOR 25MHz ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>r6</td>
<td>Min Delay Thru Transparent Latches</td>
<td>t Rd 3.3</td>
</tr>
<tr>
<td>r7</td>
<td>Min Read Pulse Width</td>
<td>t Sys - t Rd 7</td>
</tr>
<tr>
<td>r8</td>
<td>Cache Enable to Sample</td>
<td>t Smp - t Rd 0</td>
</tr>
<tr>
<td>r9</td>
<td>Read Buffer Enable to Sample</td>
<td>t Smp - t Rd 7.5</td>
</tr>
<tr>
<td>r10</td>
<td>Read Buffer Data Setup to End of Write</td>
<td>t Smp - t Rd 3</td>
</tr>
</tbody>
</table>
CALCULATING $t_{RdMin}$

t_{Smp} = 6

t_{Sys} = 12

r_6 = 3.3
r_7 = 7

r_8 = 0
r_9 = 7.5
r_{10} = 3

$MIN$ = 0
$MAX$ = 5

$t_{RdMin} = 6$
## CALCULATING $t_{Rd\ Max}$

<table>
<thead>
<tr>
<th>FORMULA</th>
<th>DESCRIPTION</th>
<th>FOR 25MHz ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>r11</td>
<td>Data Hold from End of Write</td>
<td>$t_{Smp} - t_{Rd}$ -2</td>
</tr>
<tr>
<td>r12</td>
<td>Address Hold from End of Write</td>
<td>$t_{Smp} - t_{Rd}$ -2</td>
</tr>
<tr>
<td>r18</td>
<td>Read-Read, ICache-DCache Contention</td>
<td>$t_{Sys} - t_{Rd}$ 1.6</td>
</tr>
<tr>
<td>r19</td>
<td>Read-Read, Read Buffer-DCache Contention</td>
<td>$t_{Sys} - t_{Rd}$ 3</td>
</tr>
</tbody>
</table>
RIORDAN 18

Read - Read, ICache - DCache contention - This constraint guarantees no contention between the instruction cache and the data cache on a load.

\[
\begin{align*}
\text{t Sys} - t \text{ Rd} &= t \text{IRdDe} + t \text{RamEDM} - t \text{DRdDeMax} \\
&= 1 + 1.6 - 1 \\
&= 1.6
\end{align*}
\]
WHAT IS t EDM?
(ENABLE DISABLE MISMATCH)

- Outputs Must Not Overlap
WHAT IS tEDM?

TOSHIBA TC 55417P/J - 20H

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>MIN</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>tOEE</td>
<td>Output Enable to Output in Low-Z</td>
<td>0</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>tODO</td>
<td>Output Enable to Output in High-Z</td>
<td>-</td>
<td>8</td>
<td>ns</td>
</tr>
</tbody>
</table>

- There are no Specs for Same Test Conditions
- Estimated Value for tEDM is 1.6ns Allowing for Reasonable Variations in Manufacturing
CALCULATING t RdMax

t Smp = 6

t Sys = 12

r11 = -2
r12 = -2
r18 = 1.6
r19 = 3

MAX

-2

8

MIN

t RdMax = 8

3

9
CALCULATING \( t_{Rd} \)

\[
\begin{align*}
t_{RdMin} &= 6 \\
t_{RdMax} &= 8 \\
\text{SELECT} & \quad t_{Rd} = 6
\end{align*}
\]
SELECTING THE DELAY TAPS

50 MHz OSC

2 4 6 8 10 12 14 16 18 20

Phi
Rd
Smp

6 tRd
6 tSmp
12 tSys
THREE STEP RECIPE FOR 25MHz

1. SET t Smp to 6ns
   • Guarantees Data Stable
   • Allows Parity, Tag Check, Load Align to Overlap Before PHI

2. SET t Rd to t Smp
   • UNLESS t RamHd > t RamLzMin

3. SET t Sys to t RamOe after Rd
   • Check t Sys > t RamHz + 2.5ns
THREE STEP RECIPE FOR 25MHz

3. SETTING \( t_{Sys} \) to \( t_{RamOe} \) after \( Rd \)

\[
t_{RamOe} = t_{RamDoe} + t_{RamDoeDe} + \text{Rounding} = 12 + 1 + 1 = 14
\]

\[
t_{Sys} = ((t_{Cyc}/2) + t_{Rd}) - t_{RamOe} = 20 + 6 - 14 = 12
\]
THREE STEP RECIPE FOR 25MHz

VERIFICATION

STEP 2: t RamHd MUST BE > t RamLzMin

\[
0 > 2 \quad \text{OK}
\]

STEP 3: t Sys MUST BE > t RamHzMax + 2.5

\[
12 > 8 + 2.5 \\
12 > 10.5 \quad \text{OK}
\]
WRITE REGISTER TIMING

Setup Max = r25
Hold Max = r26
CALCULATING DATA BUS HOLD TIME

- Determines Minimum Time for Which Data Must Remain Valid on the Bus After it Stops Being Driven
CALCULATING SysClk DELAY

- Specifies Max & Min Delay for Inverter Used to Generate SysClk from SysOut*
CALCULATING SysClk DELAY

\[ r_{23} = 0.5 \rightarrow MAX \rightarrow Min = 0.5 \]
\[ r_{24} = -2.5 \rightarrow \]
\[ r_{22} = 7.7 \rightarrow Max = 7.7 \]

<table>
<thead>
<tr>
<th>Part</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCT 1804A</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>FCT240A</td>
<td>1.5</td>
<td>4.8</td>
</tr>
<tr>
<td>74AS04</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>