

計算機アーキテクチャ特論 (Advanced Computer Architectures)

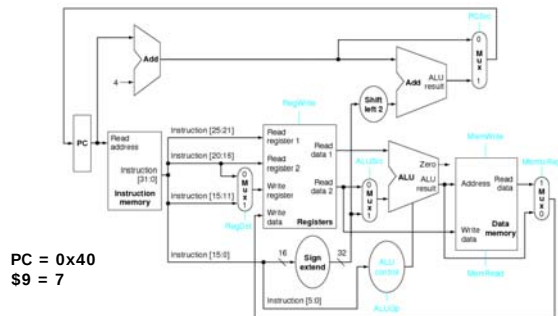
3. シングルサイクル版プロセッサの実装

吉瀬 謙二 計算工学専攻
kise_at_cs.titech.ac.jp www.arch.cs.titech.ac.jp
W831 講義室 木曜日 9:00 - 10:30

1

プロセッサのデータパス(シングル・サイクル)

op rs rt 16 bit immediate I format
addi \$t0, \$t1, -1 [addi \$8, \$9, -1]

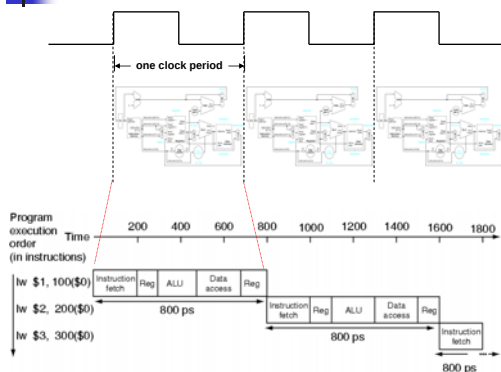


PC = 0x40
\$9 = 7

2

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

プロセッサのデータパス(シングル・サイクル)

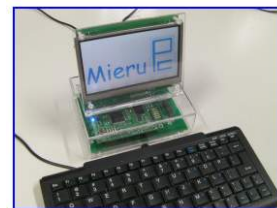


Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

3

シングルサイクル版プロセッサの実装

- Target hardware : MieruPC-2010
 - XILINX Spartan-3E XC3S250E (25万ゲート相当 FPGA)
 - 512KB SRAM



FPGA Card RC2 (ただし、50万ゲートのFPGAに変更)

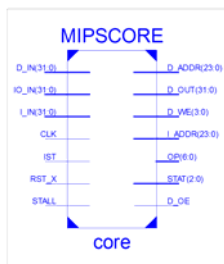
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

4

MIPSCORE (Single-Cycle Version Processor)

```
module MIPSCORE(CLK, RST_X, STALL, IST, STAT, I_ADDR, I_IN, D_ADDR, D_IN, D_OUT, D_OE, D_WE, IO_IN, OP);
    input CLK, RST_X, STALL, IST; // stall and IST(instruction stall) signal
    output reg [2:0] STAT;
    output [7:0] I_ADDR, D_ADDR;
    input [31:0] I_IN, D_IN, IO_IN;
    output [31:0] D_OUT;
    output [3:0] D_WE;
    output [6:0] D_OE;
    output [6:0] OP;
endmodule
```

- 32bit RISC Processor
- No hardware divider
 - software emulation
- No floating-point unit



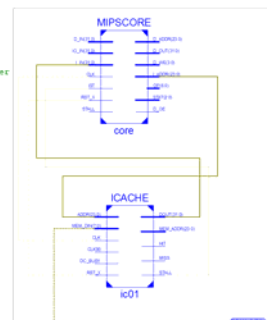
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

5

MIPSCORE & Instruction Cache

```
module ICACHE(CLK, CLK0, RST_X, MEM_ADDR, MEM_IN, ADDR, DOUT, STALL, DC_BUSY, HIT, MISS);
    input CLK, CLK0, RST_X;
    input [7:0] ADDR; // Core address
    output [7:0] DOUT; // Core data out
    input [7:0] MEM_IN; // SRAM data in
    output [7:0] MEM_ADDR; // SRAM address
    output STALL; // STALL signal
    input DC_BUSY; // Data cache busy ?
    output reg HIT, MISS; // cache hit or miss for per
endmodule
```

- simple instruction cache
- cache line : 4 byte (1 word)
- 4KB (1024 entry)
- direct mapped



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

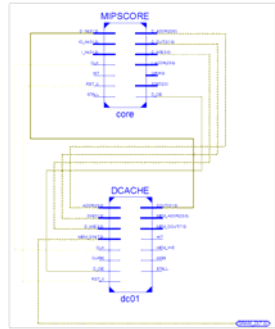
6

MIPSCORE & Data Cache

```

//-----
module DCACHE(CLK, CLK90, RST_N, D_WE, D_OE, DIN, DOUT, ADDR, MEM_DIN, MEM_DOUT, MEM_ADDR, MEM_WE,
              STALL, HIT, MISS);
    input    CLK, CLK90, RST_N;
    input [3:0] D_WE; // Core Output Enable
    input [1:0] D_OE; // Core WE
    input ['W0SD] DIN; // Core Data in
    output ['W0SD] DOUT; // Core data out
    input ['ADDR] ADDR; // Core address

```

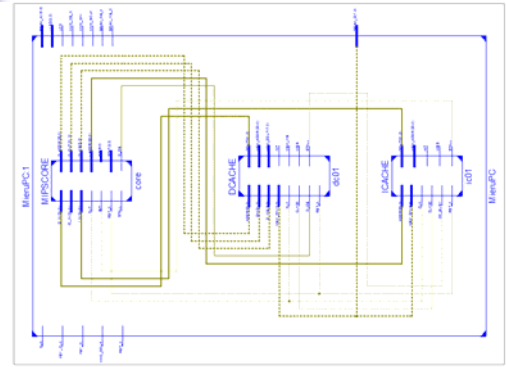


- simple data cache
- cache line : 4 byte (1 word)
- 4KB (1024 entry)
- direct mapped

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

7

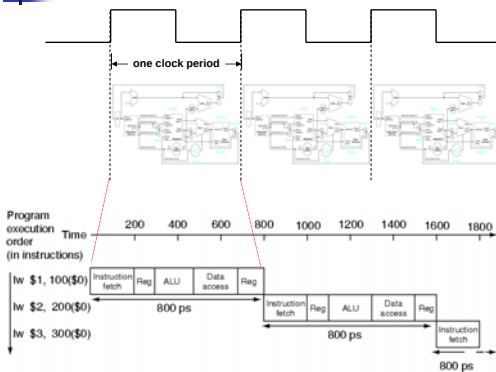
MIPSCORE & I-Cache & Data Cache



Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

8

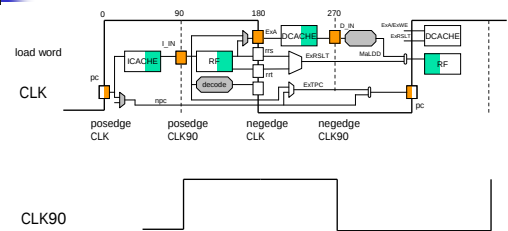
プロセッサのデータパス(シングル・サイクル)



Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

9

MIPSCORE (Single-Cycle Version Processor)

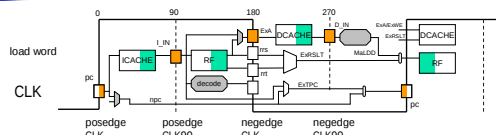


- CLK, CLK90
- positive edge, negative edge

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

10

MIPSCORE (Single-Cycle Version Processor)

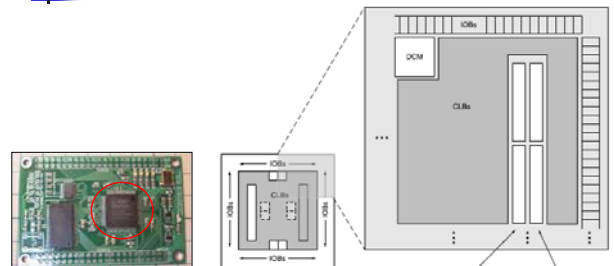


- CLK
- 18MHz, 55.5 ns
- 1/4 CLK = 13.889 ns

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

11

XILINX Spartan-3E FPGA



DCM : Digital Clock Manager
CLB : Configurable Logic Block

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

12

XILINX Spartan-3E FPGA

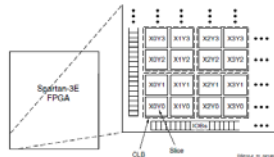


図 14: CLB のロケーション

表 9: Spartan-3E の CLB リソース

デバイス	CLB の 行数	CLB の 列数	CLB の 総数 ⁽¹⁾	スライス 数	LUT/フリップ フロップ数	専用 ロジック セル数	RAM16/ SRAM16 数	分断 RAM の ビット数
SXC3S100E	22	16	340	960	1,920	2,160	960	15,360
SXC3S150E	34	26	612	2,448	4,896	5,508	2,448	39,168
SXC3S400E	46	34	1,164	4,656	9,312	10,476	4,656	74,496
SXC3S1200E	60	46	2,168	8,672	17,344	19,512	8,672	138,752
SXC3S1400E	76	58	3,688	14,752	29,504	33,192	14,752	236,632

注 1: フラッシュ RAM は非揮発性プロビット型 DCM が CLB の範囲内に組み込まれているため (サブルーチン 2 の図 3 を参照)、CLB の総数は対称的な数の半量よりも多くなります。

Spartan-3E FPGA ファミリー: データシートより

13

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

MIPSCORE (Single-Cycle Version Processor)

Area Project Status: 01/27/2018 - 20:58:30

Project File:	testcase	Implementation State:	Programming File Generated
Module Name:	MemIO	Errors:	No Errors
Target Device:	xc3s400a-ep100	Warnings:	241 Warnings (0 new)
Product Version:	ISE 11.5	Routing Results:	All Signals Completely Routed
Design Goal:	Balanced	Timing Constraints:	All Constraints Met
Design Strategy:	Xilinx Default (unlocked)	Final Timing Score:	0 Setup, 0 Hold, 0 Component Switching Limit (0 Timing Report)

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	1,331	9,312	14%
Number of 4 input LUTs	6,425	9,312	69%
Number of occupied Slices	3,694	4,056	79%
Number of Slices containing only related logic	3,694		79%
Number of Slices containing unrelated logic	0	3,664	0%
Total Number of 4 input LUTs	7,068	9,312	76%
Number used as logic	6,041		
Number used as a router thru	647		
Number used for Dual Port RAMs	120		
Number used for SCL RAMs	268		
Number of bonded I/Os	42	66	63%
Number of RAMB16s	13	20	65%
Number of BUFMRMs	0	24	0%
Number of DCMs	2	4	50%
Average Fanout of Non-Clock Nets	3.26		

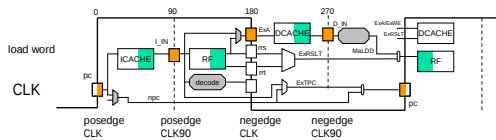
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

14

MIPSCORE (Single-Cycle Version Processor)

Slack (setup path): 0.101ns (requirement - (data path - clock path skew + uncertainty))
 Source: ic01/C1/mem0/bram_mem0.B (RAM)
 Destination: core/Exa_23 (FF)
 Requirement: 13.869ns
 Data Path Delay: 13.780ns (levels of logic = 14)
 Clock Path Skew: -0.006ns (2.011 - 2.019)
 Source Clock: FCLK90 rising at 13.888ns
 Destination Clock: FCLK falling at 27.777ns
 Clock Uncertainty: 0.000ns

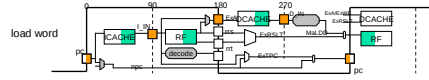
Maximum Data Path: ic01/C1/mem0/bram_mem0.B to core/Exa_23



15

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

MIPSCORE (Single-Cycle Version Processor)



Maximum Data Path: ic01/C1/mem0/bram_mem0.B to core/Exa_23

Location	Delay (ns)	Physical Resource	Logical Resource(s)	Setup (ns)	Hold (ns)	Source
RAMB16_S170_S005	0.1512	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_23752.F2	1.124	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_23752.F2	0.704	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_23752.F2	1.056	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_23752.F2	0.759	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	1.195	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.759	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	1.114	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	1.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.000	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0	0.000	0.110	core/Exa_23
Slice_24762.F1	0.110	ic01/C1/mem0/bram_mem0	ic01/C1/mem0/bram_mem0			

Operand fetch

```
98 wire [31:0] rca_t = GP$OUT0; // output0 of regfile
99 wire [31:0] rst_t = GP$OUT1; // output1 of regfile
100 reg [31:0] rca_ext; //
101 reg [*ADDB] Eka; // memory address for LB/ST
102
103 always @(negedge CLK) Eka <- rca_t['ADDB] + ((R[1_N], I_IN[15:0]));
104 always @(negedge CLK) rst <- rst_t;
105 always @(negedge CLK) rca_ext <- rst_t;
```

The diagram illustrates the operand fetch stage of a processor. It shows the internal components and their interactions during a load word operation:

- PC (Program Counter):** Provides the instruction address to the ICACHE.
- ICACHE (Instruction Cache):** Outputs the fetched instruction to the RE (Register File).
- RE (Register File):** Receives the instruction and outputs register indices (ra, rt, rd) to the decoder.
- Decoder:** Decodes the register indices and outputs control signals (rca, rst, rca_ext) to the ALU and other components.
- ALU:** Performs operations based on the ALU_OP signal and inputs from the PC and registers.
- PC (Program Counter):** Updated by the ALU result via the pc_sel signal.

The diagram also shows the connection between the ALU and the D-CACHE (Data Cache), which is used for storing or retrieving data from memory.

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

19

Memory access & write back

```

197 wire memm_out = (memm_out <= !DST_ANY) ? 0 : 0;
198 assign D_ADR0 = (memm_out) ? Eax : 0;
199 assign D_OUT = (memm_out) ? EAX_OUT : 0;
200 assign D_WE = (memm_out) ? EAX_WE : 0;
201 assign D_WE = memm_out;
202
203 wire [31:0] MALL0 = {EAX[23]} > IO_IN == 0 ? EAX[10:0] : // align here
204 (ISATTT & LD_2B) > D_IN == 0 ? EAX[10:0] : // align here
205 (ISATTT & LD_2B) && EAX[1] ? {16'h0, D_IN[31:16]} : D_IN;
206
207 wire [31:0] MALL0 = (memm_out) ? 0 : 0 // zero if not LD inst.
208 (ISOPPM <= LD) > {16'h0, MALL0[7:0]} : // extend here
209 (ISOPPM <= LD) > {16'h0, MALL0[7:0]} : // extend here
210 (ISOPPM <= LD) > {16'h0, MALL0[15:0]} : // extend here
211 (ISOPPM <= LD) > {16'h0, MALL0[15:0]} : // extend here
212
213 wire [31:0] MALL0 = (ISATTT & LD_ANY) ? MALL0 : EAX_OUT; // MALL0_val;
214
215 //*** WB, write back
216
217 GPP gpe (<CLK, <CLK, <MEM0[0], <MEM0[1], <MEM0[2], <MEM0[3], <MEM0[4], <MEM0[5], <MEM0[6], <MEM0[7], <MEM0[8], <MEM0[9], <MEM0[10], <MEM0[11], <MEM0[12], <MEM0[13], <MEM0[14], <MEM0[15], <MEM0[16], <MEM0[17], <MEM0[18], <MEM0[19], <MEM0[20], <MEM0[21], <MEM0[22], <MEM0[23], <MEM0[24], <MEM0[25], <MEM0[26], <MEM0[27], <MEM0[28], <MEM0[29], <MEM0[30], <MEM0[31], <MEM0[32], <MEM0[33], <MEM0[34], <MEM0[35], <MEM0[36], <MEM0[37], <MEM0[38], <MEM0[39], <MEM0[40], <MEM0[41], <MEM0[42], <MEM0[43], <MEM0[44], <MEM0[45], <MEM0[46], <MEM0[47], <MEM0[48], <MEM0[49], <MEM0[50], <MEM0[51], <MEM0[52], <MEM0[53], <MEM0[54], <MEM0[55], <MEM0[56], <MEM0[57], <MEM0[58], <MEM0[59], <MEM0[60], <MEM0[61], <MEM0[62], <MEM0[63], <MEM0[64], <MEM0[65], <MEM0[66], <MEM0[67], <MEM0[68], <MEM0[69], <MEM0[70], <MEM0[71], <MEM0[72], <MEM0[73], <MEM0[74], <MEM0[75], <MEM0[76], <MEM0[77], <MEM0[78], <MEM0[79], <MEM0[80], <MEM0[81], <MEM0[82], <MEM0[83], <MEM0[84], <MEM0[85], <MEM0[86], <MEM0[87], <MEM0[88], <MEM0[89], <MEM0[90], <MEM0[91], <MEM0[92], <MEM0[93], <MEM0[94], <MEM0[95], <MEM0[96], <MEM0[97], <MEM0[98], <MEM0[99], <MEM0[100], <MEM0[101], <MEM0[102], <MEM0[103], <MEM0[104], <MEM0[105], <MEM0[106], <MEM0[107], <MEM0[108], <MEM0[109], <MEM0[110], <MEM0[111], <MEM0[112], <MEM0[113], <MEM0[114], <MEM0[115], <MEM0[116], <MEM0[117], <MEM0[118], <MEM0[119], <MEM0[120], <MEM0[121], <MEM0[122], <MEM0[123], <MEM0[124], <MEM0[125], <MEM0[126], <MEM0[127], <MEM0[128], <MEM0[129], <MEM0[130], <MEM0[131], <MEM0[132], <MEM0[133], <MEM0[134], <MEM0[135], <MEM0[136], <MEM0[137], <MEM0[138], <MEM0[139], <MEM0[140], <MEM0[141], <MEM0[142], <MEM0[143], <MEM0[144], <MEM0[145], <MEM0[146], <MEM0[147], <MEM0[148], <MEM0[149], <MEM0[150], <MEM0[151], <MEM0[152], <MEM0[153], <MEM0[154], <MEM0[155], <MEM0[156], <MEM0[157], <MEM0[158], <MEM0[159], <MEM0[160], <MEM0[161], <MEM0[162], <MEM0[163], <MEM0[164], <MEM0[165], <MEM0[166], <MEM0[167], <MEM0[168], <MEM0[169], <MEM0[170], <MEM0[171], <MEM0[172], <MEM0[173], <MEM0[174], <MEM0[175], <MEM0[176], <MEM0[177], <MEM0[178], <MEM0[179], <MEM0[180], <MEM0[181], <MEM0[182], <MEM0[183], <MEM0[184], <MEM0[185], <MEM0[186], <MEM0[187], <MEM0[188], <MEM0[189], <MEM0[190], <MEM0[191], <MEM0[192], <MEM0[193], <MEM0[194], <MEM0[195], <MEM0[196], <MEM0[197], <MEM0[198], <MEM0[199], <MEM0[200], <MEM0[201], <MEM0[202], <MEM0[203], <MEM0[204], <MEM0[205], <MEM0[206], <MEM0[207], <MEM0[208], <MEM0[209], <MEM0[210], <MEM0[211], <MEM0[212], <MEM0[213], <MEM0[214], <MEM0[215], <MEM0[216], <MEM0[217], <MEM0[218], <MEM0[219], <MEM0[220], <MEM0[221], <MEM0[222], <MEM0[223], <MEM0[224], <MEM0[225], <MEM0[226], <MEM0[227], <MEM0[228], <MEM0[229], <MEM0[230], <MEM0[231], <MEM0[232], <MEM0[233], <MEM0[234], <MEM0[235], <MEM0[236], <MEM0[237], <MEM0[238], <MEM0[239], <MEM0[240], <MEM0[241], <MEM0[242], <MEM0[243], <MEM0[244], <MEM0[245], <MEM0[246], <MEM0[247], <MEM0[248], <MEM0[249], <MEM0[250], <MEM0[251], <MEM0[252], <MEM0[253], <MEM0[254], <MEM0[255], <MEM0[256], <MEM0[257], <MEM0[258], <MEM0[259], <MEM0[260], <MEM0[261], <MEM0[262], <MEM0[263], <MEM0[264], <MEM0[265], <MEM0[266], <MEM0[267], <MEM0[268], <MEM0[269], <MEM0[270], <MEM0[271], <MEM0[272], <MEM0[273], <MEM0[274], <MEM0[275], <MEM0[276], <MEM0[277], <MEM0[278], <MEM0[279], <MEM0[280], <MEM0[281], <MEM0[282], <MEM0[283], <MEM0[284], <MEM0[285], <MEM0[286], <MEM0[287], <MEM0[288], <MEM0[289], <MEM0[290], <MEM0[291], <MEM0[292], <MEM0[293], <MEM0[294], <MEM0[295], <MEM0[296], <MEM0[297], <MEM0[298], <MEM0[299], <MEM0[300], <MEM0[301], <MEM0[302], <MEM0[303], <MEM0[304], <MEM0[305], <MEM0[306], <MEM0[307], <MEM0[308], <MEM0[309], <MEM0[310], <MEM0[311], <MEM0[312], <MEM0[313], <MEM0[314], <MEM0[315], <MEM0[316], <MEM0[317], <MEM0[318], <MEM0[319], <MEM0[320], <MEM0[321], <MEM0[322], <MEM0[323], <MEM0[324], <MEM0[325], <MEM0[326], <MEM0[327], <MEM0[328], <MEM0[329], <MEM0[330], <MEM0[331], <MEM0[332], <MEM0[333], <MEM0[334], <MEM0[335], <MEM0[336], <MEM0[337], <MEM0[338], <MEM0[339], <MEM0[340], <MEM0[341], <MEM0[342], <MEM0[343], <MEM0[344], <MEM0[345], <MEM0[346], <MEM0[347], <MEM0[348], <MEM0[349], <MEM0[350], <MEM0[351], <MEM0[352], <MEM0[353], <MEM0[354], <MEM0[355], <MEM0[356], <MEM0[357], <MEM0[358], <MEM0
```

Whetstone Benchmark, Version 2.1 (Language: C)
 The number of runs through the benchmark: 299998, runnins...
 Execution ends
 Initial values of the variables used in the benchmark:
 Int_Glob: 5(5), Bool_Glob: 1(1)
 Ch_1_Glob: A(1), Ch_2_Glob: 8(8)
 Arr_1_Glob(8): 7(7), Arr_2_Glob(8)(7): 299919(299919)
 Ptr_Glob-> Pen_Comp: 152268
 Discor: 9(9), Enum_Comp: 2(2)
 Int_Comp: 17(17), Str_Comp:
 WAX MIPS rating = 13.356

clk stall	18992	748	8812	735
shift miss rate :	18173	71	993	
dshift miss rate :	1699	1	999	

runnins at 18 MHz 13 MIPS

load word

CLK

posedge CLK

posedge CLK90

negedge CLK

negedge CLK90

PC

ICACHE

DCACHE

RF

CLK90

- CLK, CLK90
 - positive edge, negative edge
- **18MHz, 13.3 VAX MIPS**

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

22



アナウンス

- 講義スライド, 講義スケジュール
 - www.arch.cs.titech.ac.jp

23

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005