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

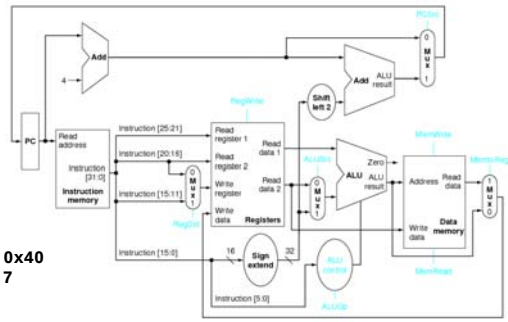
吉瀬 謙二 計算工学専攻
kise _at_ cs.titech.ac.jp www.arch.cs.titech.ac.jp
W831 講義室 木曜日 9:00 – 10:30

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

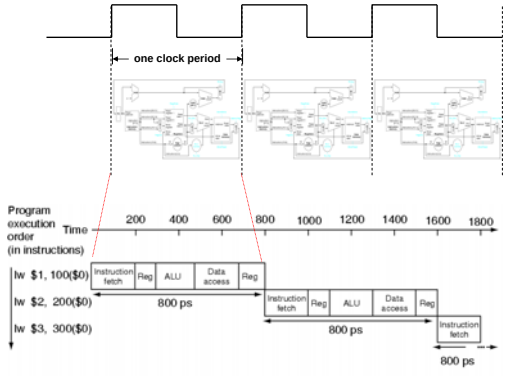
I format

addi $t0, $t1, -1 **[addi $8, $9, -1]**



PC = 0x40
$9 = 7

プロセッサのデータパス（シングル・サイクル）



シングルサイクル版プロセッサの実装

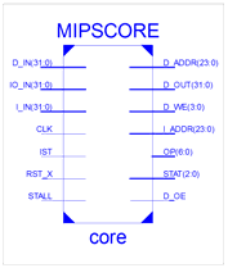
- Target hardware：MieruPC-2010
  - XILINX Spartan-3E XC3S250E (25万ゲート相当 FPGA)
  - 512KB SRAM

FPGA Card RC2 (ただし，50万ゲートのFPGAに変更)

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 [`ADDR]   I_ADDR, D_ADDR;
    input  [31:0]    I_IN, D_IN, IO_IN;
    output [31:0]    D_OUT;
    output [3:0]     D_WE;
    output           D_OE;
    output [6:0]     OP;
```

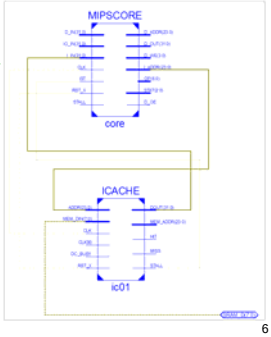
- 32bit RISC Processor
- No hardware divider
  - software emulation
- No floating-point unit



MIPSCORE & Instruction Cache

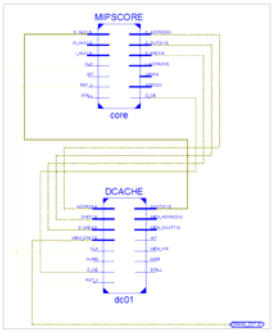
```
/*****************************************************************************/
module ICACHE(CLK, CLKSO, RST_X, MEM_ADDR, MEM_DIN, ADDR, DOUT, STALL, DC_BUSY, HIT, MISS);
    input          CLK, CLKSO, RST_X;
    input  [`ADDR] ADDR;      // Core address
    output [`WORD] DOUT;      // Core data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

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



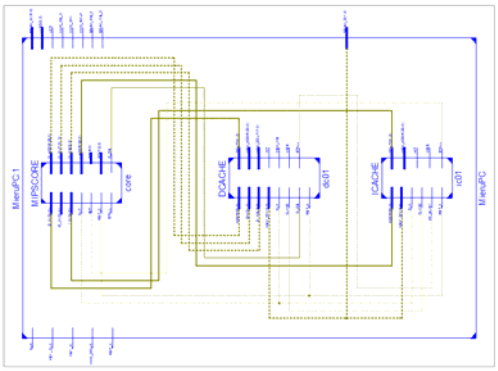
MIPSCORE & Data Cache

```
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          D_OE;     // Core Output Enable
  input    [3:0] D_WE;     // Core WE
  input  [`WORD] DIN;      // Core Data in
  output [`WORD] DOUT;     // Core data out
  input  [`ADDR] ADDR;     // Core address
```

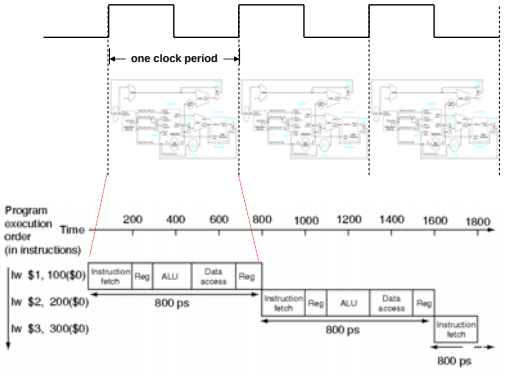


- simple data cache
- cache line : 4 byte (1 word)
- 4KB (1024 entry)
- direct mapped

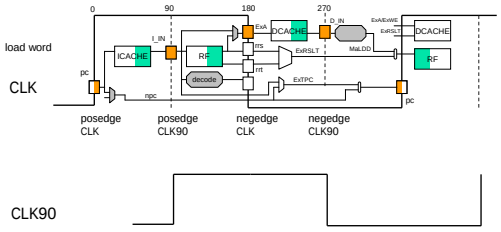
MIPSCORE & I-Cache & Data Cache



プロセッサのデータパス（シングル・サイクル）

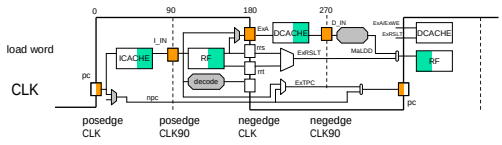


MIPSCORE (Single-Cycle Version Processor)



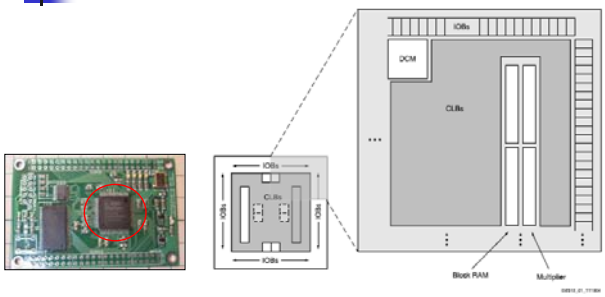
- CLK, CLK90
  - positive edge, negative edge

MIPSCORE (Single-Cycle Version Processor)



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

XILINX Spartan-3E FPGA



DCM：Digital Clock Manager
CLB：Configurable Logic Block

Spartan-3E FPGA ファミリ：データシート より

XILINX Spartan-3E FPGA

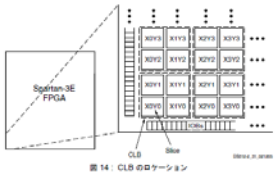


図 14 : CLB のロケーション

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	等価ロジックセル数	RAM16/SRL16 数	分散 RAM のビット数
XC3S100E	22	16	340	960	1,920	2,160	960	15,360
XC3S250E	34	26	612	2,448	4,896	5,508	2,448	39,168
XC3S500E	46	34	1,164	4,656	9,312	10,476	4,656	74,496
XC3S1200E	60	46	2,168	8,672	17,344	19,512	8,672	138,752
XC3S1600E	76	58	3,688	14,752	29,504	33,192	14,752	236,032

メモ
1. ブロック RAM や乗算器ブロックおよび DCM が CLB の配列内に組み込まれているため (モジュール 1 の図 1 を参照)、CLB の総数は行と列の積の乗算値よりも少なくなります。

Spartan-3E FPGA ファミリ: データシート より

MIPSCORE (Single-Cycle Version Processor)

fpga Project Status (10/27/2010 - 20:58:35)

Project File:	fpga.ise	**Implementation State:**	Programming File Generated
Module Name:	MemuPC	• **Errors:**	No Errors
Target Device:	xc3s500e-4vq100	• **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)

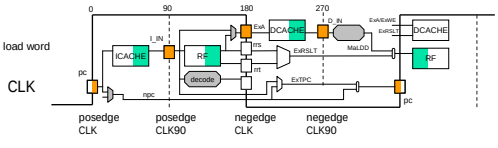
Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	1,329	9,312	14%	
Number of 4 input LUTs	6,425	9,312	68%	
Number of occupied Slices	3,694	4,656	79%	
Number of Slices containing only related logic	3,694	3,694	100%	
Number of Slices containing unrelated logic	0	3,694	0%	
Total Number of 4 input LUTs	7,066	9,312	75%	
Number used as logic	6,041			
Number used as a route-thru	641			
Number used for Dual Port RAMs	128			
Number used for 32x1 RAMs	256			
Number of bonded IOBs	42	66	63%	
Number of RAMB16s	13	20	65%	
Number of BUFG/BUFGMUXs	3	24	12%	
Number of DCMs	2	4	50%	
Average Fanout of Non-Clock Nets	3.26			

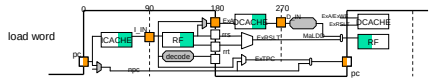
MIPSCORE (Single-Cycle Version Processor)

```
--------------------------------------------------------------------------------
Slack (setup path):     0.101ns (requirement - (data path - clock path skew + uncertainty))
  Source:               ic01/C1/mem0/Mram_mem2.B (RAM)
  Destination:          core/ExA_23 (FF)
  Requirement:          13.889ns
  Data Path Delay:      13.780ns (Levels of Logic = 14)
  Clock Path Skew:      -0.008ns (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/Mram_mem2.B to core/ExA_23
```



MIPSCORE (Single-Cycle Version Processor)

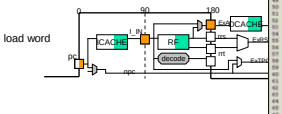


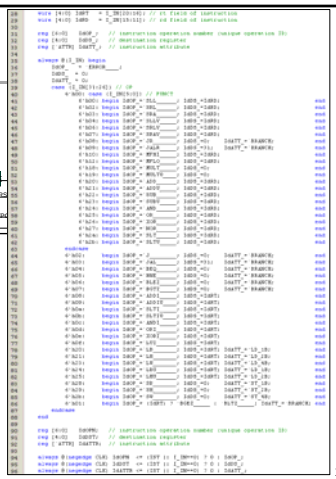

```
Maximum Data Path: ic01/C1/mem0/Mram_mem2.B to core/ExA_23
  Location              Delay type         Delay(ns)  Physical Resource
                                                      Logical Resource(s)
  ----------------------------------------------------------------------
  RAMB16_X1Y8.DOB5      Tbcko              2.812      ic01/C1/mem0/Mram_mem2
                                                      ic01/C1/mem0/Mram_mem2.B
  SLICE_X53Y52.F2       net (fanout=2)     1.324      ic01/DO<23>
  SLICE_X53Y52.X        Tilo               0.704      CORE_I_IN<23>
                                                      ic01/DOUT<23>1
  SLICE_X50Y65.F3       net (fanout=64)    1.686      CORE_I_IN<23>
  SLICE_X50Y65.X        Tilo               0.759      core/N15
                                                      core/gpr/Mram_r6.SLICEM_F
  SLICE_X44Y63.F3       net (fanout=1)     1.196      core/N15
  SLICE_X44Y63.X        Tilo               0.759      core/Result__AUX_1_29
                                                      core/gpr/DOUT0<2>1
  SLICE_X43Y39.F3       net (fanout=2)     1.314      core/GPROUT0<2>
  SLICE_X43Y39.COUT     Topcyf             1.162      core/ExA<1>
                                                      core/Madd_ExA_add0000_lut<2>
                                                      core/Madd_ExA_add0000_cy<2>
                                                      core/Madd_ExA_add0000_cy<3>
  SLICE_X43Y40.CIN      net (fanout=1)     0.000      core/Madd_ExA_add0000_cy<3>
  SLICE_X43Y40.COUT     Tbyp               0.118      core/ExA<4>
                                                      core/Madd_ExA_add0000_cy<4>
                                                      core/Madd_ExA_add0000_cy<5>
  SLICE_X43Y41.CIN      net (fanout=1)     0.000      core/Madd_ExA_add0000_cy<5>
  SLICE_X43Y41.COUT     Tbyp               0.118      core/ExA<6>
                                                      core/Madd_ExA_add0000_cy<6>
                                                      core/Madd_ExA_add0000_cy<7>
  SLICE_X43Y42.CIN      net (fanout=1)     0.000      core/Madd_ExA_add0000_cy<7>
  SLICE_X43Y42.COUT     Tbyp               0.118      core/ExA<8>
                                                      core/Madd_ExA_add0000_cy<8>
                                                      core/Madd_ExA_add0000_cy<9>
  SLICE_X43Y43.CIN      net (fanout=1)     0.000      core/Madd_ExA_add0000_cy<9>
  SLICE_X43Y43.COUT     Tbyp               0.118      core/ExA<10>
                                                      core/Madd_ExA_add0000_cy<10>
                                                      core/Madd_ExA_add0000_cy<11>
  SLICE_X43Y44.CIN      net (fanout=1)     0.000      core/Madd_ExA_add0000_cy<11>
  SLICE_X43Y44.COUT     Tbyp               0.118      core/ExA<12>
                                                      core/Madd_ExA_add0000_cy<12>
                                                      core/Madd_ExA_add0000_cy<13>
  SLICE_X43Y45.CIN      net (fanout=1)     0.000      core/Madd_ExA_add0000_cy<13>
  SLICE_X43Y45.COUT     Tbyp               0.118      core/ExA<14>
                                                      core/Madd_ExA_add0000_cy<14>
                                                      core/Madd_ExA_add0000_cy<15>
  SLICE_X43Y46.CIN      net (fanout=1)     0.000      core/Madd_ExA_add0000_cy<15>
  SLICE_X43Y46.COUT     Tbyp               0.118      core/ExA<16>
                                                      core/Madd_ExA_add0000_cy<16>
                                                      core/Madd_ExA_add0000_cy<17>
  SLICE_X43Y47.CIN      net (fanout=1)     0.000      core/Madd_ExA_add0000_cy<17>
  SLICE_X43Y47.COUT     Tbyp               0.118      core/ExA<18>
                                                      core/Madd_ExA_add0000_cy<18>
                                                      core/Madd_ExA_add0000_cy<19>
  SLICE_X43Y48.CIN      net (fanout=1)     0.000      core/Madd_ExA_add0000_cy<19>
  SLICE_X43Y48.COUT     Tbyp               0.118      core/ExA<20>
                                                      core/Madd_ExA_add0000_cy<20>
                                                      core/Madd_ExA_add0000_cy<21>
  SLICE_X43Y49.CIN      net (fanout=1)     0.000      core/Madd_ExA_add0000_cy<21>
  SLICE_X43Y49.CLK      Tcinck             1.002      core/ExA<22>
                                                      core/Madd_ExA_add0000_cy<22>
                                                      core/Madd_ExA_add0000_xor<23>
                                                      core/ExA_23
  ----------------------------------------------------------------------
  Total                                    13.780ns (8.240ns logic, 5.520ns route)
                                                    (59.8% logic, 40.1% route)
```


MIPSCORE (Single-Cycle Version Processor)

- Video

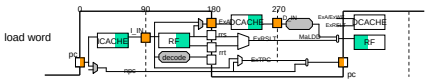
Decode



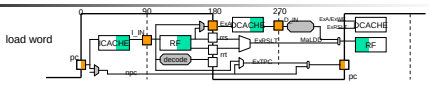


Operand fetch

```
98   wire [31:0] rrs_t = GPROUT0;  // output0 of regfile
99   wire [31:0] rrt_t = GPROUT1;  // output1 of regfile
100  reg  [31:0] rrs, rrt;         //
101  reg [`ADDR] ExA;              // memory address for LD/ST
102
103  always @(negedge CLK) ExA <= rrs_t[`ADDR] + ((8{I_IN[15]}}, I_IN[15:0]});
104  always @(negedge CLK) rrs <= rrs_t;
105  always @(negedge CLK) rrt <= rrt_t;
```



Memory access & write back

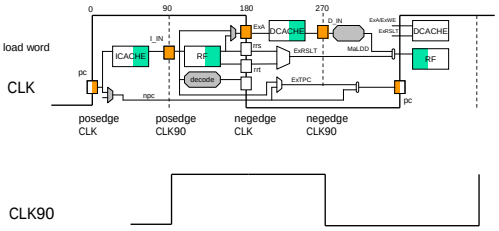



```
187  wire mem_oe = (IdATTR & `LDST_ANY) ? 1 : 0;
188  assign D_ADDR = (mem_oe) ? ExA    : 0;
189  assign D_OUT  = (mem_oe) ? ExRSLT : 0;
190  assign D_WE   = (mem_oe) ? ExWE   : 0;
191  assign D_OE   =  mem_oe;
192
193  wire [31:0] MaALIGN =  (ExA[2])                    ? IO_IN >> (8*ExA[1:0]) :  ///// align here
194                         (IdATTR & `LD_1B)           ? D_IN  >> (8*ExA[1:0]) :
195                         ((IdATTR & `LD_2B) && ExA[1]) ? {16'd0, D_IN[31:16]}  : D_IN;
196
197  wire [31:0] MaLDD = (mem_oe==0) ? 0 : // zero if not LD inst.
198                      (IdOPN==`LB_____) ? {{24{MaALIGN[ 7]}}, MaALIGN[ 7:0]} : /////extend here
199                      (IdOPN==`LBU____) ? { 24'd0           , MaALIGN[ 7:0]} :
200                      (IdOPN==`LH_____) ? {{16{MaALIGN[15]}}, MaALIGN[15:0]} :
201                      (IdOPN==`LHU____) ? { 16'd0           , MaALIGN[15:0]} : MaALIGN;
202
203  wire [31:0] MaRSLT = (IdATTR & `LD_ANY) ? MaLDD : ExRSLT; //MaWb_rslt;
204
205  /***** WB, write back                                                                    *****/
206  /*****************************************************************************************/
207  GPR gpr(.CLK(CLK), .REGNUM0(I_IN[25:21]), .REGNUM1(I_IN[20:16]), .REGNUM2(IdDST),
208          .DIN0(MaRSLT), .WE0(~STALL), .DOUT0(GPROUT0), .DOUT1(GPROUT1));
209
210  always @( posedge CLK or negedge RST_X ) begin  ///// update pc, hi, lo register
211      if(!RST_X) {pc, hi, lo} <= 0;
212      else if(~STALL) begin
213          pc <= (ExC) ? ExTPC : pc + 4; // If branch taken, uses taken PC else uses pc+4
214          if(IdOPN == `MULT____) {hi, lo} <= RRS_S * RRT_S; // very slow!!
215          if(IdOPN == `MULTU___) {hi, lo} <= RRS_U * RRT_U; // very slow!!
216      end
217  end
```


MIPSCORE (Single-Cycle Version Processor)

- Video

MIPSCORE (Single-Cycle Version Processor)



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

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