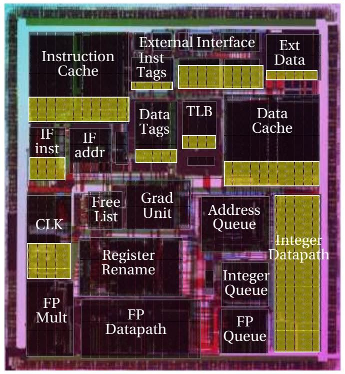
The MIPS R10000 Processor

Ruud van der Pas

SiliconGraphics

European High Performance Computing Team

R10000 Physical Layout



R10000 Design Goals

- *Very high application performance*
- *Broad system application range*
- *Less sensitivity to code order*
- *Multi–vendor mainstream CMOS*
- *Single chip implementation*

Compute Bottlenecks

- *Program flow changes* *(branches, jumps)*
- *Register dependencies*
- *Availability of data or code* *(bandwidth issue)*
- *Resource conflicts* *(caches, system bus)*

Keys to Superscalar Effectiveness:

- *Must exploit instruction parallelism*
- *Must hide memory reference latencies*

The ANDES Architecture

Architecture with **N**on–Sequential
Dynamic **E**xecution **S**cheduling

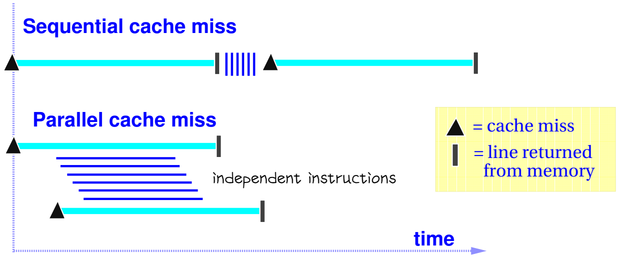
- ***Decoupled "data flow" architecture***
  - Not a traditional pipelined approach
  - Extensive use of queues
- ***Effective use of superscalar resources***
  - High issue bandwidth
  - Aggressive execution unit designs
- ***Extreme memory latency tolerance***

R10000: Main Features

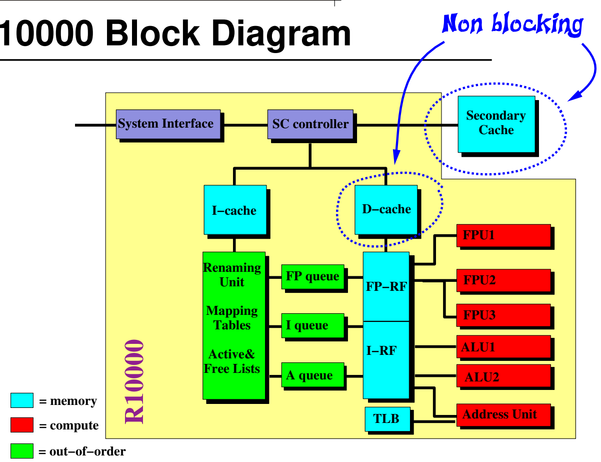
- *Supports the 64–bit MIPS IV ISA*
- *4–way superscalar*
- *Five seperate execution units*
- *2 floating point results/cycle*
- *4–way deep speculative execution of branches*
- *Out–of–order execution (32 instruction window)*
- *Precise exception handling*
- *Two user–accessible event counters*

R10000: Reduces Memory Latency

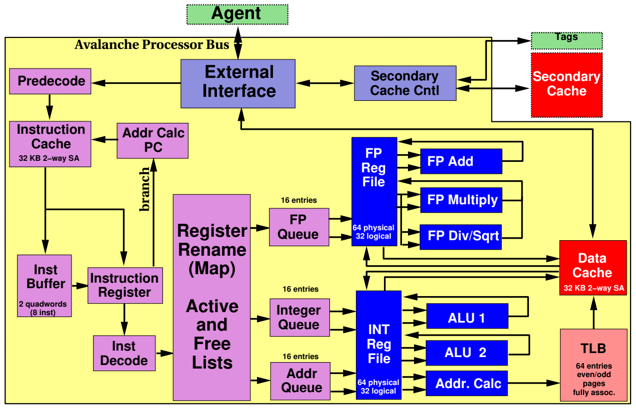
- *Two–way set associative non–blocking caches*
  - *Up to 4 outstanding memory requests*
  - *Prefetching of data*



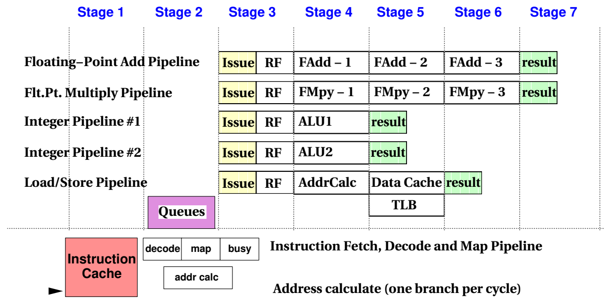
R10000 Block Diagram



R10000 Block Diagram



R10000 Data Flow Concept



Out–of–Order Execution

	Assembly Code		Physical Operation	Rename	Unit	Cycle	Comments
1	DSLL	r3,r2,2	p9 <–p2 shift left 2	r3=p9	ALU1	1A	
2	LW	r4,0x8(r3)	p10<–mem(p9+8)	r4=p10	L/ST	1B	*true dependency "r3" on inst #1*
3	ADD1	r5,r2,0x34	p11<–p2+'34'	r5=p11	ALU2	1C	
4	SUB	r3,r1,r5	p12<–p1–p11	r3=p12	ALU1	1D	*output dependency "r3" on inst #1 anti dependency "r3" on inst #2*
5	XORI	r2,r1,0xFF	p13<–p1 xor 'FF'	r2=p13	ALU2	2A	
6	BEQ	r4,r2,label	branch if p10=p13	none	ALU1	2B	*conditional branch (predict taken)*
7	NOP	–	no operation	none	ALU1	2C	*delay slot (instr after branch)*
8	MULT	r3,r4					*(not decoded if branch is taken)*
9	MFLO	r2,....					
....							
label:							
10	AND	r3,r1,r2	p14<–p1 and p13	r3=p14	ALU2	3A	
11	LB	r4,0(r3)	p15<–mem(p14+0)	r4=p15	L/ST	3B	

Out–of–Order Execution (cont)

	cycle–1	cycle–2	cycle–3	cycle–4	cycle–5	cycle–6	cycle–7	cycle–8
1 DSLL	**dec**	issue	**ALU1**	write				
2 LW	**dec**	*wait*	issue	**Address**	**DCache**	write		
3 ADDI	**dec**	issue	**ALU2**	write	—	—		
4 SUB	**dec**	*wait*	issue	**ALU1**	write	—		
5 XORI		**dec**	issue	**ALU2**	write	—		
6 BEQ		**dec**	*wait*	*wait*	issue	**ALU1**	(no wrt)	
7 NOP		**dec**	ready	issue	**ALU1**	(no wrt)	—	
10 AND			**dec**	issue	**ALU2**	write	—	
11 LB			**dec**	*wait*	issue	**Address**	**DCache**	write

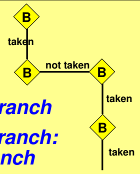
Instruction graduates if all previous have grad.

Instructions must wait to graduate in order.

Up to 4 instructions graduate per cycle.

Branch Prediction

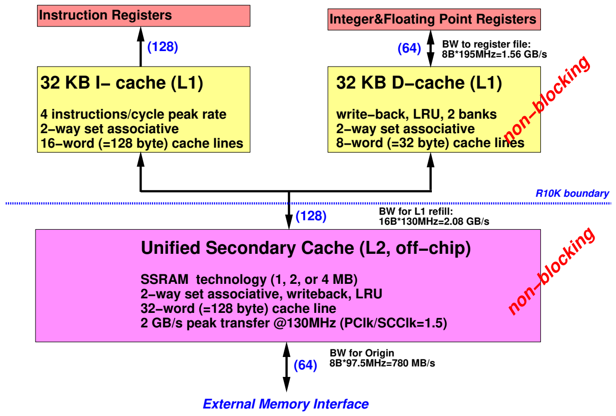
- Speculates on all branches
- Up to four nested branch predictions
- 512–entry, 2–bit prediction table
- Shadow copy of register map made for each branch
- Single–cycle misprediction recovery for any branch:
  - shadow copy of reg map made for each branch
  - register map prior to prediction is restored
  - dest registers of aborted instructions returned to free list



➡ *Prevents that the CPU stalls*

➡ *Generates early cache misses*
(to overlap execution)

Cache Architecture



Instruction Latencies

INTEGER UNITS			Latency	Repeat Rate
ALU1	add, sub, logic ops, shift, branches		1	1
ALU2	add, sub, logic ops			
	multiply	(32/64 bit)	6/10	6/10
	divide	(32/64 bit)	35/67	35/67
ADDRESS UNIT				
load int/FP			2/3	1
store				1
FLOATING POINT UNITS				
FPU1	add, sub, compare, convert		2	1
FPU2	multiply		2	1
	multiply–add (MADD)		4	1
FPU3	divide	(32/64 bit)	12/19	14/21
	sqrt	(32/64 bit)	18/33	20/35

Observations

Instructions

- *Take advantage of the parallelism in the instruction set*

a = b+c can run in parallel with **d=e*f**

Memory Hierarchy

- *Re–use data where possible*
- *L1 cache is relatively small, so difficult to store all data*
- *This is easier to achieve for the L2 cache*
- *Prefetch allows to overlap memory access and compute part*

Use stride 1 where possible!

Performance: Recommendations

- **Try to achieve overlap of:**

  Compute and memory reference

  *Functional units (e.g. perform * and + in paralle)*

- **Minimize data copies**

  If you can't avoid them, add floating point operations to it

  Often these are then "for free"

- **De–vectorize your code**

  Watch out for scalars promoted to arrays in order to please a vector system ...

The Event Counters

- **The R10000 has two on–chip event counters**
- **They are accessible through:**
  - The IRIX "perfex" command
  - The profiling environment
  - A procedural interface (Fortran and C)
- **Two events out of a selection of 32 can be monitored**
- **This is a great tool to analyze performance!**

The Perfex Command

Syntax:

```
% perfex [–e e0] [–e e1] [–a] [–mp] [–y] [–s] [–x] file [options]
```

Perfex options:

e0 and e1	event identifiers
–a	multiplex over all events
–mp	report counts/thread as well as totals
–y	gives timing estimates
–s	start/stop counting on USR1/USR2 signal
–x	count on exception level as well

(note: perfex writes to stderr)

The Perfex Command (cont)

event 0

0	Cycles
1	Issued instructions
2	Issued loads
3	Issued stores
4	Issued store conditionals
5	Failed store conditionals
6	Decoded branches
7	Quadwords written back from scache
8	Correctable scache data array ECC errors
9	Primary instruction cache misses
10	Secondary instruction cache misses
11	Instruction misprediction from scache way prediction table
12	External interventions
13	External invalidations
14	Virtual coherency conditions
15	Graduated instructions

event 1

16	Cycles
17	Graduated instructions
18	Graduated loads
19	Graduated stores
20	Graduated store conditionals
21	Graduated floating point instructions
22	Quadwords written back from primary data cache
23	TLB misses
24	Mispredicted branches
25	Primary data cache misses
26	Secondary data cache misses
27	Data misprediction from scache way prediction table
28	External intervention hits in scache
29	External invalidation hits in scache
30	Store/prefetch exclusive to clean block in scache
31	Store/prefetch exclusive to shared block in scache

The Perfex Command (cont)

Examples:

```
% perfex -e 2 -e 25 ./swp
 KREP =            100 N =              300 B(N/2,N/2) =    22.00000
 Memory requirement =   0.6866455     MB
 Elapsed time =   2.4454899E-03 (   290.5070     Mflop/s)

 Issued loads ..................................        13847619
 Primary data cache misses......................         2310349
```

```
% perfex -e 3 -e 26 ./swp
 KREP =            100 N =              300 B(N/2,N/2) =    22.00000
 Memory requirement =   0.6866455     MB
 Elapsed time =   2.4720700E-03 (   287.3835     Mflop/s)

 Issued stores..................................         9176152
 Secondary data cache misses....................            9029
```


Example Perfex Output

Cntr 0

```
 0 Cycles..............................................................  1645481936
 1 Issued instructions.................................................   677976352
 2 Issued loads........................................................   111412576
 3 Issued stores.......................................................    45085648
 4 Issued store conditionals...........................................           0
 5 Failed store conditionals...........................................           0
 6 Decoded branches....................................................    52196528
 7 Quadwords written back from scache..................................    61794304
 8 Correctable scache data array ECC errors............................           0
 9 Primary instruction cache misses....................................        8560
10 Secondary instruction cache misses..................................         304
11 Instruction misprediction from scache way prediction table..........         272
12 External interventions..............................................        6144
13 External invalidations..............................................       10032
14 Virtual coherency conditions........................................           0
15 Graduated instructions..............................................   371427616
```

Cntr 1

```
16 Cycles..............................................................  1645481936
17 Graduated instructions..............................................   400535904
18 Graduated loads.....................................................    90474112
19 Graduated stores....................................................    34776112
20 Graduated store conditionals........................................           0
21 Graduated floating point instructions...............................    28292480
22 Quadwords written back from primary data cache......................    32386400
23 TLB misses..........................................................     5687456
24 Mispredicted branches...............................................      410064
25 Primary data cache misses...........................................    16330160
26 Secondary data cache misses.........................................     7708944
27 Data misprediction from scache way prediction table.................      663648
28 External intervention hits in scache................................        6144
29 External invalidation hits in scache................................        6864
30 Store/prefetch exclusive to clean block in scache...................     7582256
31 Store/prefetch exclusive to shared block in scache..................        8144
```


Example (continued)

Event Counter Name	Counter Value	Typical Time (sec)	Minimum Time (sec)	Maximum Time (sec)
0 Cycles	1639802080	8.366337	8.366337	8.366337
16 Cycles	1639802080	8.366337	8.366337	8.366337
26 Secondary data cache misses	7736432	2.920580	1.909429	3.248837
23 TLB misses	5693808	1.978017	1.978017	1.978017
7 Quadwords written back from scache	61712384	1.973562	1.305834	1.973562
25 Primary data cache misses	16368384	0.752445	0.235504	0.752445
22 Quadwords written back from primary data cache	32385280	0.636139	0.518825	0.735278
2 Issued loads	109918560	0.560809	0.560809	0.560809
18 Graduated loads	88890736	0.453524	0.453524	0.453524
6 Decoded branches	52497360	0.267844	0.267844	0.267844
3 Issued stores	43923616	0.224100	0.224100	0.224100
19 Graduated stores	33430240	0.170562	0.170562	0.170562
21 Graduated floating point instructions	28371152	0.144751	0.072375	7.527040
30 Store/prefetch exclusive to clean block in scache	7545984	0.038500	0.038500	0.038500
24 Mispredicted branches	417440	0.003024	0.001363	0.011118
9 Primary instruction cache misses	8272	0.000761	0.000238	0.000761
10 Secondary instruction cache misses	768	0.000290	0.000190	0.000323
31 Store/prefetch exclusive to shared block in scache	15168	0.000077	0.000077	0.000077
1 Issued instructions	673476960	0.000000	0.000000	3.436107
4 Issued store conditionals	0	0.000000	0.000000	0.000000
5 Failed store conditionals	0	0.000000	0.000000	0.000000
8 Correctable scache data array ECC errors	0	0.000000	0.000000	0.000000
11 Instruction misprediction from scache way prediction table	432	0.000000	0.000000	0.000002
12 External interventions	6288	0.000000	0.000000	0.000000
13 External invalidations	9360	0.000000	0.000000	0.000000
14 Virtual coherency conditions	0	0.000000	0.000000	0.000000
15 Graduated instructions	364303776	0.000000	0.000000	1.858693
17 Graduated instructions	392675440	0.000000	0.000000	2.003446
20 Graduated store conditionals	0	0.000000	0.000000	0.000000
27 Data misprediction from scache way prediction table	679120	0.000000	0.000000	0.003465
28 External intervention hits in scache	6288	0.000000	0.000000	0.000000
29 External invalidation hits in scache	5952	0.000000	0.000000	0.000000

Example (continued)

```
Statistics
=========================================================================================
Graduated instructions/cycle...............................................    0.222163
Graduated floating point instructions/cycle................................    0.017302
Graduated loads & stores/cycle.............................................    0.074595
Graduated loads & stores/floating point instruction........................    5.422486
Mispredicted branches/Decoded branches.....................................    0.007952
Graduated loads/Issued loads...............................................    0.808696
Graduated stores/Issued stores.............................................    0.761099
Data mispredict/Data scache hits...........................................    0.078675
Instruction mispredict/Instruction scache hits.............................    0.057569
L1 Cache Line Reuse........................................................    6.473003
L2 Cache Line Reuse........................................................    1.115754
L1 Data Cache Hit Rate.....................................................    0.866185
L2 Data Cache Hit Rate.....................................................    0.527355
Time accessing memory/Total time...........................................    0.750045
L1--L2 bandwidth used (MB/s, average per process)..........................  124.541093
Memory bandwidth used (MB/s, average per process)..........................  236.383187
MFLOPS (average per process)...............................................    3.391108
```

The Cost Table

```
% perfex -t

                                                          Costs for IP27 processor
                                                                  Typical     Minimum     Maximum
   Event Counter Name                                                Cost        Cost        Cost
=================================================================================================
 0 Cycles.......................................................  1.00 clks   1.00 clks   1.00 clks
 1 Issued instructions..........................................  0.00 clks   0.00 clks   1.00 clks
 2 Issued loads.................................................  1.00 clks   1.00 clks   1.00 clks
 3 Issued stores................................................  1.00 clks   1.00 clks   1.00 clks
 4 Issued store conditionals....................................  1.00 clks   1.00 clks   1.00 clks
 5 Failed store conditionals....................................  1.00 clks   1.00 clks   1.00 clks
 6 Decoded branches.............................................  1.00 clks   1.00 clks   1.00 clks
 7 Quadwords written back from scache...........................  6.40 clks   4.23 clks   6.40 clks
 8 Correctable scache data array ECC errors.....................  0.00 clks   0.00 clks   1.00 clks
 9 Primary instruction cache misses............................. 18.02 clks   5.63 clks  18.02 clks
10 Secondary instruction cache misses........................... 75.50 clks  49.36 clks  84.00 clks
11 Instruction misprediction from scache way prediction table...  0.00 clks   0.00 clks   1.00 clks
12 External interventions.......................................  0.00 clks   0.00 clks   0.00 clks
13 External invalidations.......................................  0.00 clks   0.00 clks   0.00 clks
14 Virtual coherency conditions.................................  0.00 clks   0.00 clks   0.00 clks
15 Graduated instructions.......................................  0.00 clks   0.00 clks   1.00 clks
16 Cycles.......................................................  1.00 clks   1.00 clks   1.00 clks
17 Graduated instructions.......................................  0.00 clks   0.00 clks   1.00 clks
18 Graduated loads..............................................  1.00 clks   1.00 clks   1.00 clks
19 Graduated stores.............................................  1.00 clks   1.00 clks   1.00 clks
20 Graduated store conditionals.................................  1.00 clks   1.00 clks   1.00 clks
21 Graduated floating point instructions........................  1.00 clks   0.50 clks  52.00 clks
22 Quadwords written back from primary data cache...............  3.85 clks   3.14 clks   4.45 clks
23 TLB misses................................................... 68.09 clks  68.09 clks  68.09 clks
24 Mispredicted branches........................................  1.42 clks   0.64 clks   5.22 clks
25 Primary data cache misses....................................  9.01 clks   2.82 clks   9.01 clks
26 Secondary data cache misses.................................. 75.50 clks  49.36 clks  84.00 clks
27 Data misprediction from scache way prediction table..........  0.00 clks   0.00 clks   1.00 clks
28 External intervention hits in scache.........................  0.00 clks   0.00 clks   0.00 clks
29 External invalidation hits in scache.........................  0.00 clks   0.00 clks   0.00 clks
30 Store/prefetch exclusive to clean block in scache............  1.00 clks   1.00 clks   1.00 clks
31 Store/prefetch exclusive to shared block in scache...........  1.00 clks   1.00 clks   1.00 clks
```


Counters: Profiling Environment

- ***The profiler gives full access to the event counters***
- ***Counter values are measured at the procedural level***
- ***An example (with TLB misses measured) now follows***

Profiling Environment – Example

```
% ssrun -tlb_hwc adi2
 Time:        7.887 seconds
 Checksum:   5.6160428338E+06
% prof adi2.tlb_hwc.18520
-------------------------------------------------------------------------------
Profile listing generated Wed Jan 15 19:27:52 1997
    with:       prof adi2.tlb_hwc.18520
-------------------------------------------------------------------------------

        Counter                : TLB misses
        Counter overflow value: 257
        Total numer of ovfls   : 21181
        CPU                    : R10000
        FPU                    : R10010
        Clock                  : 195.0MHz
        Number of CPUs         : 1


-------------------------------------------------------------------------------
  -p[rocedures] using counter overflow.
  Sorted in descending order by the number of overflows in each procedure.
  Unexecuted procedures are excluded.
-------------------------------------------------------------------------------

  overflows(%)  cum overflows(%)      procedure (dso:file)

  21169( 99.9)      21169( 99.9)         zsweep (adi2:adi2.f)
      4(  0.0)      21173(100.0)         ysweep (adi2:adi2.f)
      4(  0.0)      21177(100.0)         xsweep (adi2:adi2.f)
      4(  0.0)      21181(100.0)       fake_adi (adi2:adi2.f)

  21181(100.0)                          TOTAL
```


Perfex Procedural Interface

C SYNOPSIS

```
int start_counters ( int e0, int e1 );
int read_counters ( int e0, long long *c0, int e1, long long *c1);
int print_counters ( int e0, long long c0, int e1, long long c1);
```

FORTRAN SYNOPSIS

```
INTEGER*8 c0, c1
INTEGER*4 function start_counters ( e0, e1 )
INTEGER*4 function read_counters ( e0, c0, e1, c1 )
INTEGER*4 function print_counters ( e0, c0, e1, c1 )
```

The event specifiers e0 and e1 can also be set (or overruled) at runtime by two environment variables: T5_EVENT0 and T5_EVENT1

Notes:

- Link with -lperfex to resolve these functions
- The counter values c0 and c1 are 64 bit integers

R10000 Summary

Designed for optimal application performance:

- *Four–way superscalar*
- *Out–of–order execution*
- *High performance caches*
- *Non–blocking memory operations*
- *Dataflow concept to hide memory latency*