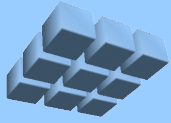


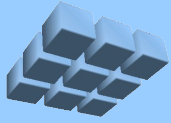
# **Extending Processor Models in the *Cmpware* *CMP-DK***

Steven A. Guccione  
*Cmpware, Inc.*

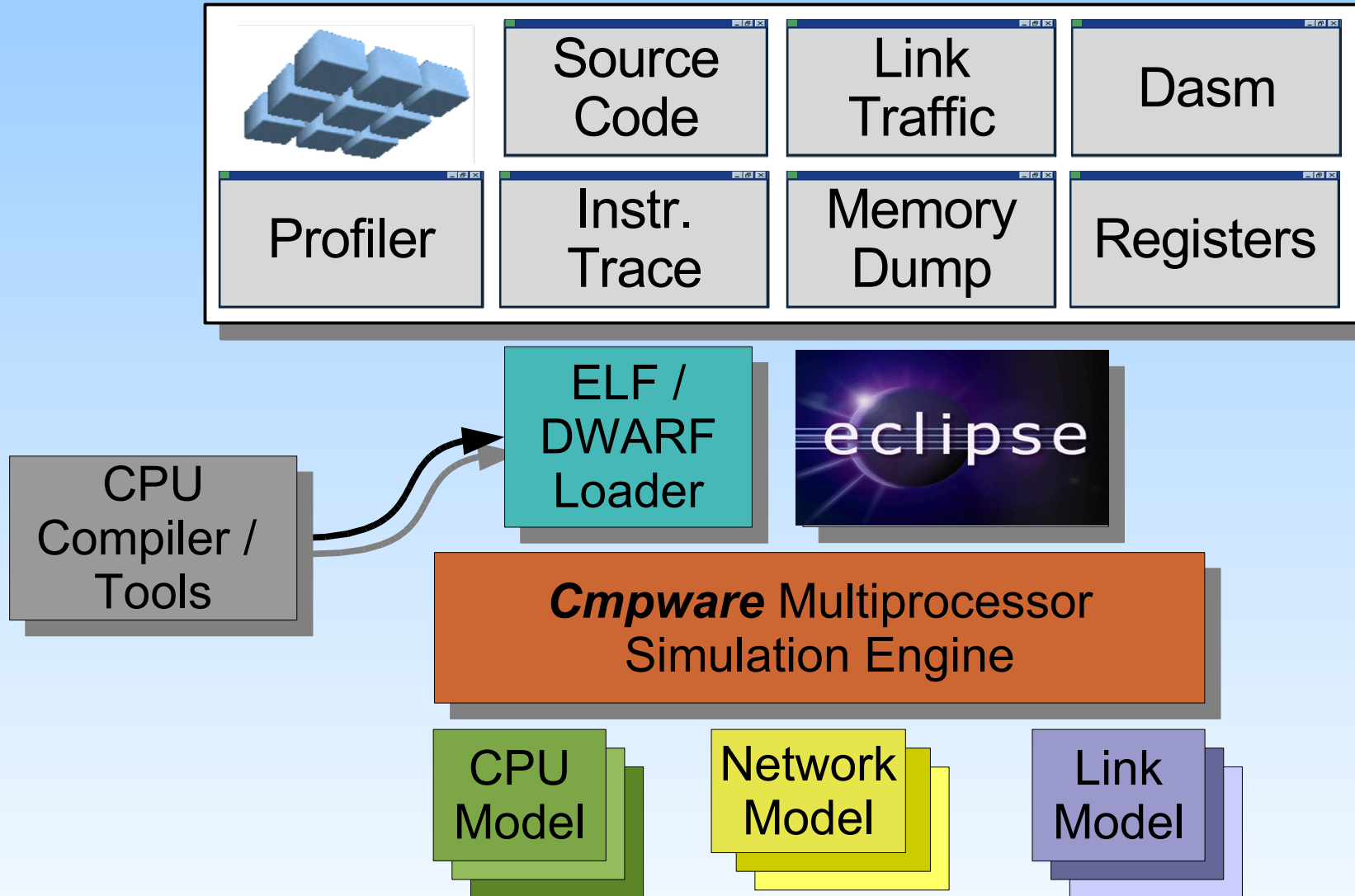


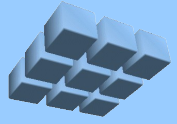
# The *Cmpware CMP-DK*

- Multiprocessor / multicore software and architecture development environment
- A powerful interactive software development
- Fast, flexible modeling
- CPU model 'plugs in' to provide rich set of displays and tools
- Integrates with existing compilers (gcc)
- Eclipse based



# The *Cmpware* CMP-DK

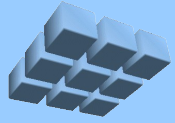




# Extending the *Processor Model*

- New models can *'inherit'* existing models
- New instructions easily added
- Other functionality easily added, too
- A new model with new instruction in approximately 5 lines of code (;)
- Models can be mixed (heterogeneous)

**==> A powerful way to explore customized processors / multiprocessors**



# The *MIPS32EX* Extended Model

```
public class MIPS32EX extends MIPS32 {

public static void main(String[] args) {
    AutoModel.main(new MIPS32EX(), args);
} /* end main() */

/** The constructor */
public MIPS32EX() {
    defineName("MIPS32EX");
    defineInstructions(new Instructions);
    reset();
} /* end MIPS32EX() */

/** The XNOR instruction */
private final class XNOR implements Function {
    public void exec(Processor p) {
        r[rd] = ~(r[rs] ^ r[rt]);
    } /* end exec() */
} /* end class XNOR */

/** The XNOR instruction decode information */
private final static Decode newi_decode = Decode.primary(op, 0x18);

/** The new XNOR instruction definition */
private final Instruction newInstructions[] = {
    new Instruction("xnor", regToReg, newi_decode, new XNOR())
};

} /* end class MIPS32EX */
```

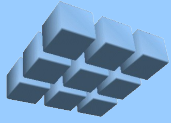
*MIPS32EX*  
'extends' *MIPS32*

Add new  
instruction(s)

Define new XNOR  
instruction

Define XNOR  
decode

The new  
instruction  
definition(s)



# Model-Driven Tools

- Assembler and disassembler automatically extracted from model data
  - Assists in verifying new models
  - Tools always in sync with architecture
  - Can be integrated into other tools
- 'Free' tool support for extension instructions
- Example:
  - *XNOR* instruction added to *MIPS32*
  - Produces *MIPS32EX* model assembler / disassembler



# Model-Based Assembler / Disassembler

```
Command Prompt
C:\Users\Cmpware\Extend>type Test.asm
///
/// MIPS32 Test
/// Copyright (c) 2007 Cmpware, Inc. All Rights Reserved.
///
addi r1, r2, 5
xor r4, r4, r1
xnor r5, r5, r1

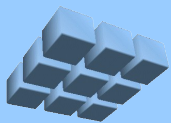
C:\Users\Cmpware\Extend>java com.cmpware.cmp.models.MIPS32 -asm Test.asm Test.out
20410005 --> addi r1, r2, 5
00812026 --> xor r4, r4, r1
Error at line 9: xnor r5, r5, r1. Exiting.
Unknown operation: xnor.

C:\Users\Cmpware\Extend>java com.cmpware.cmp.models.MIPS32EX -asm Test.asm Test.out
20410005 --> addi r1, r2, 5
00812026 --> xor r4, r4, r1
60A12800 --> xnor r5, r5, r1
3 instructions processed. (0 #define / 4 comment / 2 blank lines).

C:\Users\Cmpware\Extend>_
```

MIPS32 has no  
XNOR instruction

MIPS32EX has  
XNOR instruction



# A MIPS32EX + MIPS32 Design

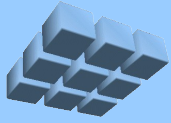
The screenshot shows the Cmpware Eclipse SDK interface. On the left, a register table lists registers r0 through r31 with their current values. On the right, a 'CMP Array' window displays a 4x4 grid of colored squares representing memory or data. Two callouts, 'MIPS32' and 'MIPS32EX', point to specific elements in the array.

r[n]	Name	Value
0	r0	0
1	r1	5
2	r2	0
3	r3	0
4	r4	5
5	r5	ffffffa
6	r6	0
7	r7	0
8	r8	0
9	r9	
10	r10	
11	r11	
12	r12	
13	r13	
14	r14	
15	r15	0
16	r16	0
17	r17	0
18	r18	0
19	r19	0
20	r20	0
21	r21	0
22	r22	0
23	r23	0
24	r24	0
25	r25	0
26	r26	0
27	r27	0
28	r28	4000
29	r29	7ff0
30	r30	7ff0
31	r31	0

The 'CMP Array' window shows a 4x4 grid of squares. The top row contains four squares: blue, orange, blue, orange. The second row contains orange, blue, orange, blue. The third row contains blue, orange, blue, orange. The bottom row contains orange, blue, orange, blue. A callout labeled 'MIPS32' points to the top-left blue square. A callout labeled 'MIPS32EX' points to the top-right orange square.

Status window text:  
Welcome to the Cmpware CMP-DK version 2.2.0 (special).  
MIPS32(0,0) selected.  
Execution step (cycle = 1)  
MIPS32EX(0,1) selected.  
Warning: file C:\Users\Cmpware\Extend\Test.out not in ELF format. Loading as binary.





# The *MIPS32* Processor

Registers:

r[n]	Name	Value
0	r0	0
1	r1	5
2	r2	0
3	r3	0
4	r4	5
5	r5	0
6	r6	0
7	r7	0
8	r8	0
9	r9	0
10	r10	0
11	r11	0
12	r12	0
13	r13	0
14	r14	0
15	r15	0
16	r16	0
17	r17	0
18	r18	0
19	r19	0
20	r20	0
21	r21	0
22	r22	0
23	r23	0
24	r24	0
25	r25	0
26	r26	0
27	r27	0
28	r28	4000
29	r29	7ff0
30	r30	7ff0
31	r31	0

Disassembly:

```
00000000: 20410005 addi r1, r2, 5
00000004: 00812026 xor r4, r4, r1
>00000008: 60a12800
0000000c: 00000000 nop
00000010: 00000000 nop
00000014: 00000000 nop
00000018: 00000000 nop
0000001c: 00000000 nop
00000020: 00000000 nop
00000024: 00000000 nop
00000028: 00000000 nop
0000002c: 00000000 nop
00000030: 00000000 nop
00000034: 00000000 nop
00000038: 00000000 nop
0000003c: 00000000 nop
00000040: 00000000 nop
00000044: 00000000 nop
```

Register Values  
(Note r4 = xor(r4, r1))

Disassembly  
(note no XNOR)

Status: MIPS32(0,2) selected.  
MIPS32EX(0,1) selected.  
MIPS32(0,0) selected.

# The *MIPS32EX* Processor

The screenshot displays the Cmpware - Eclipse SDK interface. On the left, a table shows register values. On the right, a disassembly window shows assembly code. Two callout boxes highlight specific details: 'Register Values (Note r5 = xnor(r5, r1))' and 'Disassembly (note XNOR)'.

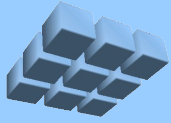
r[n]	Name	Value
0	r0	0
1	r1	5
2	r2	0
3	r3	0
4	r4	5
5	r5	fffffffa
6	r6	0
7	r7	0
8	r8	0
9	r9	0
10	r10	0
11	r11	0
12	r12	0
13	r13	0
14	r14	0
15	r15	0
16	r16	0
17	r17	0
18	r18	0
19	r19	0
20	r20	0
21	r21	0
22	r22	0
23	r23	0
24	r24	0
25	r25	0
26	r26	0
27	r27	0
28	r28	4000
29	r29	7ff0
30	r30	7ff0
31	r31	0

```
00000000: 20410005 addi r1, r2, 5
00000004: 00812026 xor r4, r4, r1
00000008: 60a12800 xnor r5, r5, r1
>0000000c: 00000000 nop
00000010: 00000000 nop
00000014: 00000000 nop
00000018: 00000000 nop
0000001c: 00000000 nop
00000020: 00000000 nop
00000024: 00000000 nop
00000028: 00000000 nop
0000002c: 00000000 nop
00000030: 00000000 nop
00000034: 00000000 nop
00000038: 00000000 nop
0000003c: 00000000 nop
00000040: 00000000 nop
00000044: 00000000 nop
00000048: 00000000 nop
0000004c: 00000000 nop
00000050: 00000000 nop
00000054: 00000000 nop
00000058: 00000000 nop
0000005c: 00000000 nop
00000060: 00000000 nop
00000064: 00000000 nop
00000068: 00000000 nop
0000006c: 00000000 nop
00000070: 00000000 nop
00000074: 00000000 nop
00000078: 00000000 nop
0000007c: 00000000 nop
00000080: 00000000 nop
00000084: 00000000 nop
00000088: 00000000 nop
```

Register Values  
(Note r5 = xnor(r5, r1))

Disassembly  
(note XNOR)

Status Power Meter MpMon  
MIPS32(0,2) selected.  
MIPS32EX(0,1) selected.  
MIPS32(0,0) selected.  
MIPS32EX(0,1) selected.



# Conclusions

- Processor models for the *Cmpware CMP-DK* easily extended
- Extended models provide:
  - Assembler
  - Disassembler
  - IDE integration

**==> *Customized processors and tools easily produced and modified***