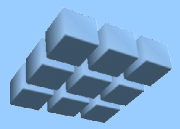
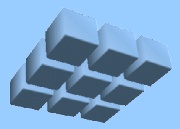


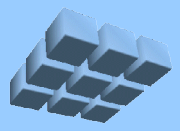
Programming Configurable Multiprocessors

Steven A. Guccione
Cmpware, Inc.



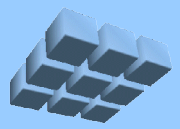
Overview

- Configurable Multiprocessing (CMP)
- An FIR Filter example
- The *Cmpware* IDE
- Conclusions



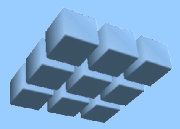
Configurable Multiprocessing

- Small, simple processors
  - Processor + compiler a “black box”
  - Not a processor + compiler design endeavor
- High-bandwidth interconnection network
  - Not shared cache
  - Not single shared bus
- Resembles hardware design
  - High bandwidth communication
  - Parallel data flow



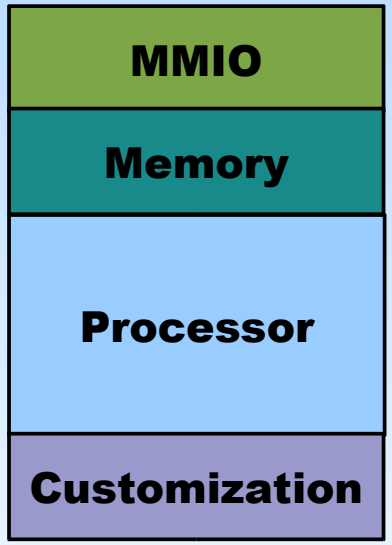
CMP Programming

- Thousands of processors on an ASIC
- Hundreds of processors on an FPGA
- Programmed in high level languages (HLLs) such as 'C' or Java
- Very high levels of raw performance: millions of MIPS
- Huge power / performace advantage over uniprocessors



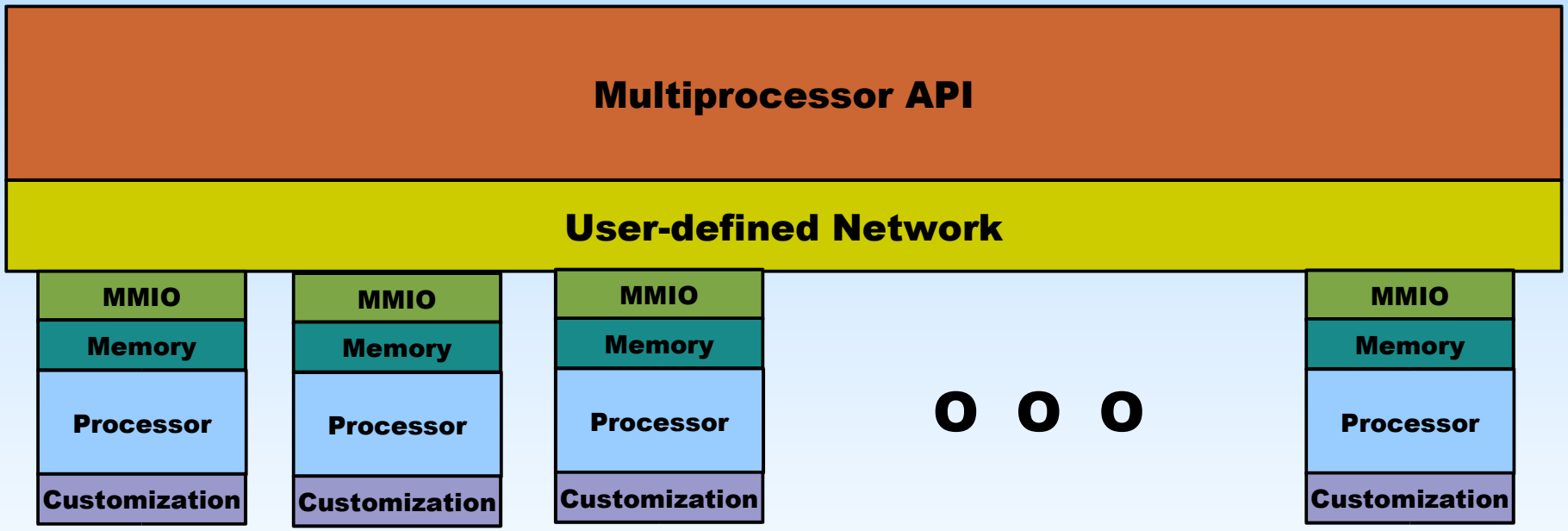
The Cmpware Processor Model

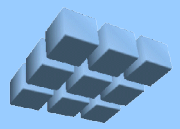
- The Cmpware Processor model heirarchy:
  - ***Customization:*** Processor-specific customizations
  - ***Processor:*** standard processor simulation machinery
  - ***Memory:*** Standard memory model
  - ***MMIO:*** Memory Mapped IO



The Cmpware Multiprocessor Model

- Network access via Memory Mapped IO
- Does not require modifications to the processor or compiler

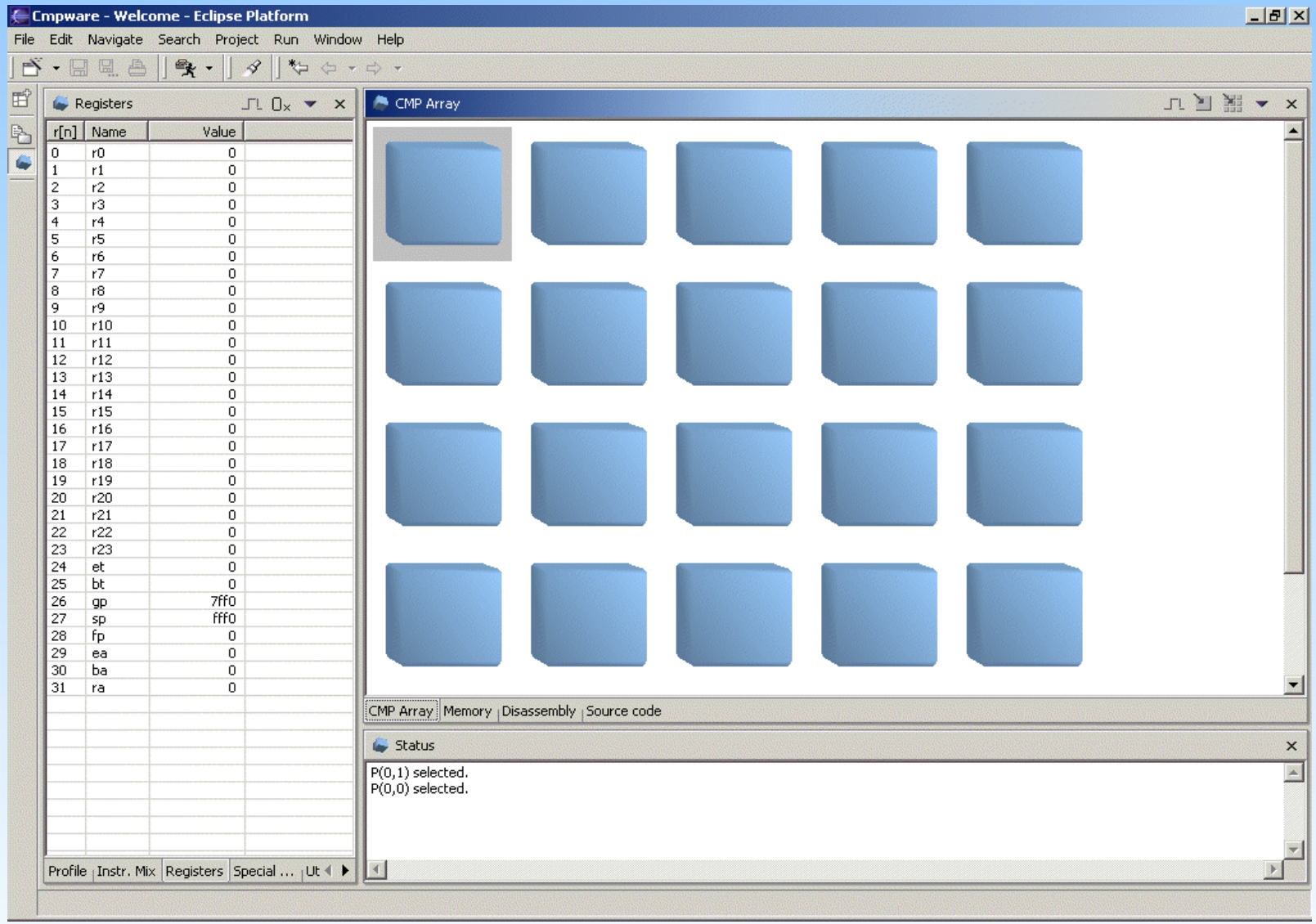




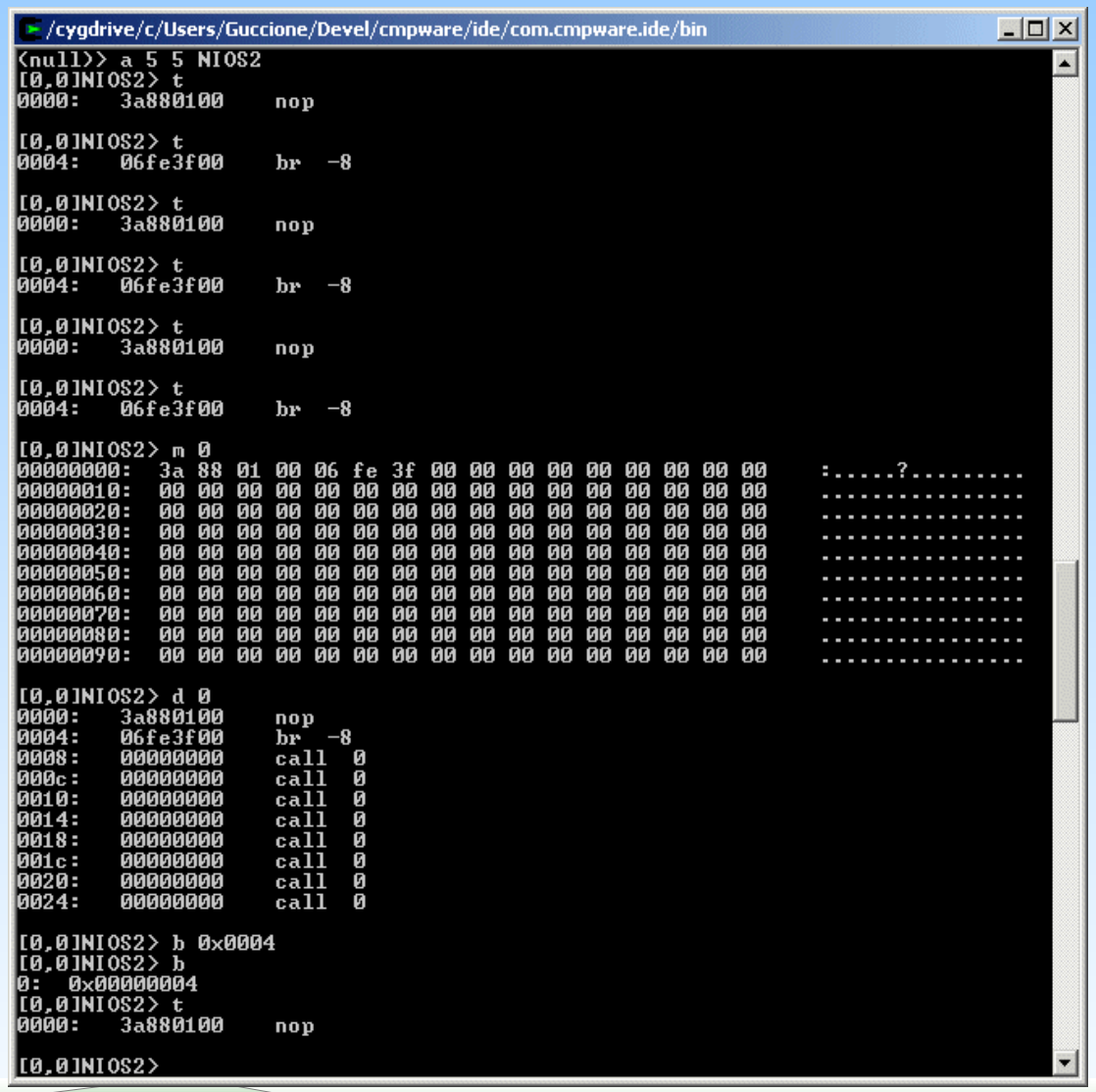
Cmpware SDK

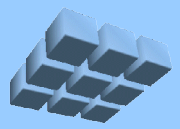
- “Pluggable” processor models
  - Run at approx. 2M cycles per second
  - Provide a variety of run-time information
- Homogeneous / heterogeneous supported
- User-definable interconnect network
- “Hardwired” nodes
- *Eclipse* IDE or command line interface
- *ProcGen* processor model generator

The Cmpware Eclipse IDE



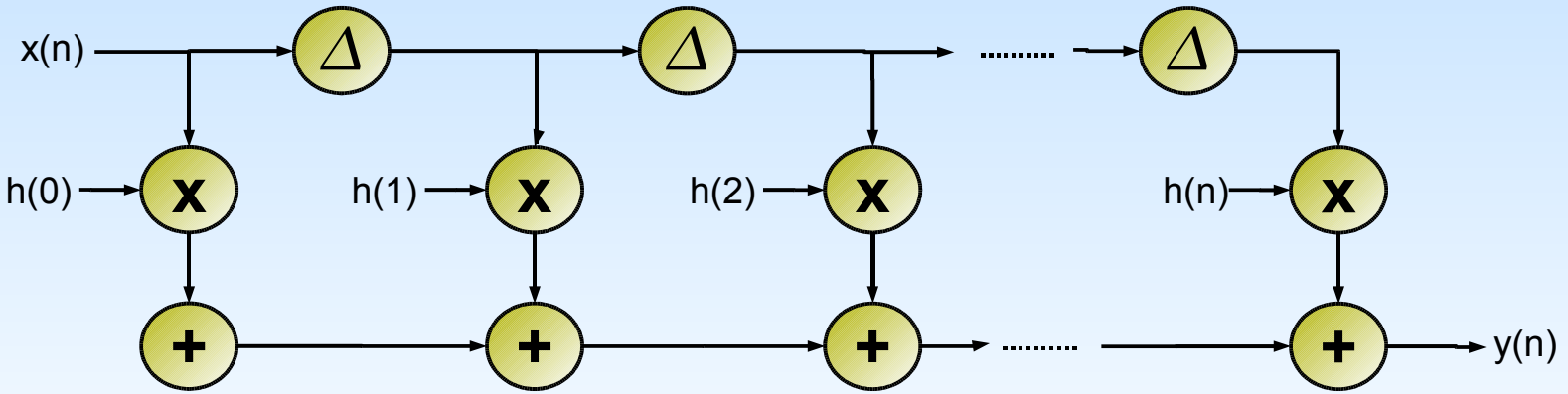
The Cmpware Command Line Interface





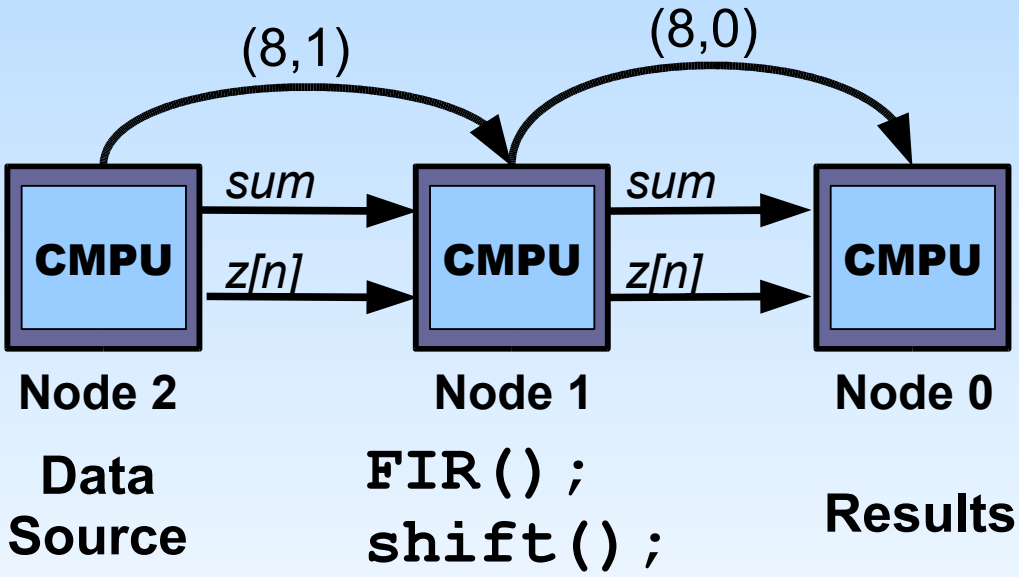
Example: FIR Filter

- Finite Impulse Response (FIR) filter
- Common Digital Signal Processing filter
- Often implemented in hardware and software
- Various methods of parallelizing



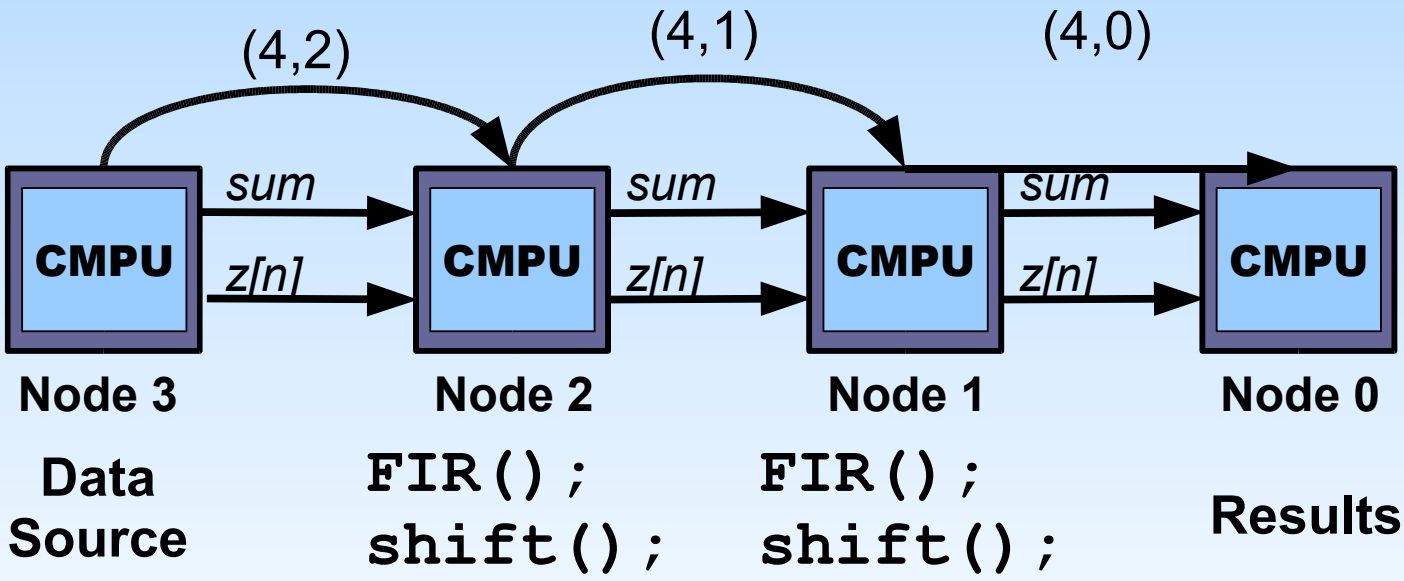
FIR Filter Implementation

- Three nodes: data source, processor, consumer
- Single 8-tap FIR



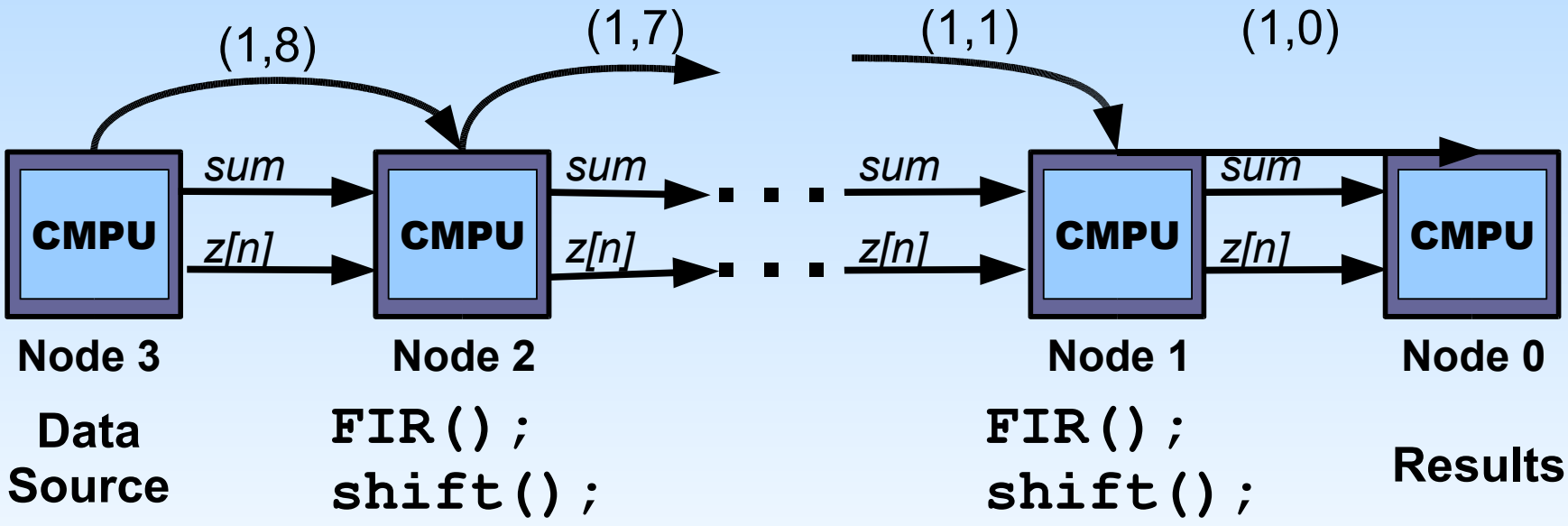
FIR Filter Implementation (cont.)

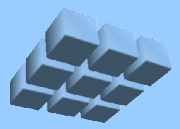
- Four nodes: data source, processor 1, processor 2, consumer
- 4x2 8-tap FIR



FIR Filter Implementation (cont.)

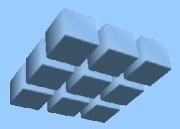
- 8 nodes: data source, 8 processors, consumer
- 8x1 8-tap FIR





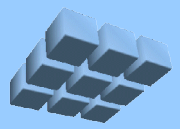
Example: FIR Filter

```
/*
**  The FIR filter main program.
*/

void  _start(void) {
   int  node;
   int  ntaps;

   /* Get the parameters */
   node = *west;
   ntaps = *west;

   /* Send the parameters to the next node */
   *east = (node-1);
   *east = ntaps;

   for (;;) {
      *east = FIR(ntaps, *west);
      *east = shift(ntaps, *west);
      }  /* end for() */

   }  /* end _start() */
```



Example: FIR Filter (cont.)

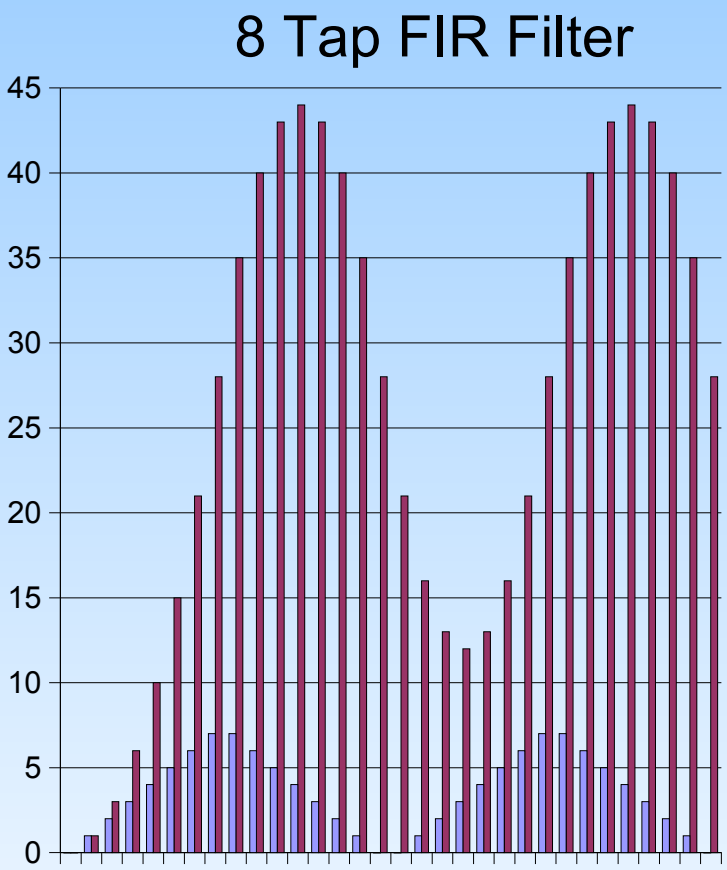
```
/*
**  This method computes the FIR.
**
**  @param  ntaps  The number of taps in the
**          FIR filter.
**
**  @param  sum  The partial sum into the FIR.
**
**  @return  This method returns the FIR result.
**
*/

int FIR(int  ntaps, int sum) {
   int  i;

   for (i=0; i<ntaps; i++)
      sum += h[i] * z[i];

   return (sum);

   }  /* end FIR() */
```

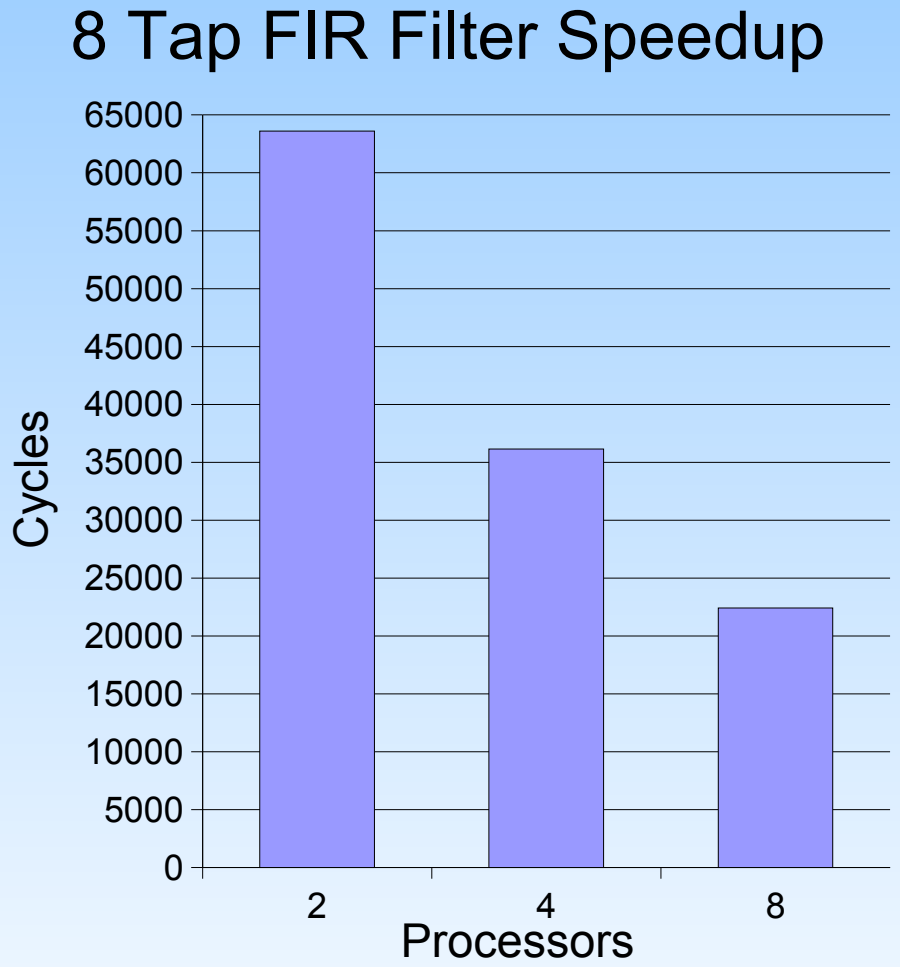


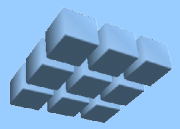
Example: FIR Filter (cont.)

```
int shift(int ntaps, int zIn) {
   int  i, zOut;

   /* Save the last value (being shifted out) */
   zOut = z[ntaps-1];

   /* Shift the delay line */
   for (i=ntaps-2; i>=0; i--)
      z[i+1] = z[i];

   /* Add in the new shifted in value */
   z[0] = zIn;

   return (zOut);

   }  /* end shift() */
```

FIR Filter Speedup

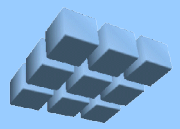






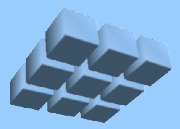
Dynamic Power / Performance Tradeoffs

- Processors in computation dynamically selectable
- Power / performace traded with a single function call parameter
- Useful in managing power consumption / battery life
- Flexibility not available in other architectures



Conclusions

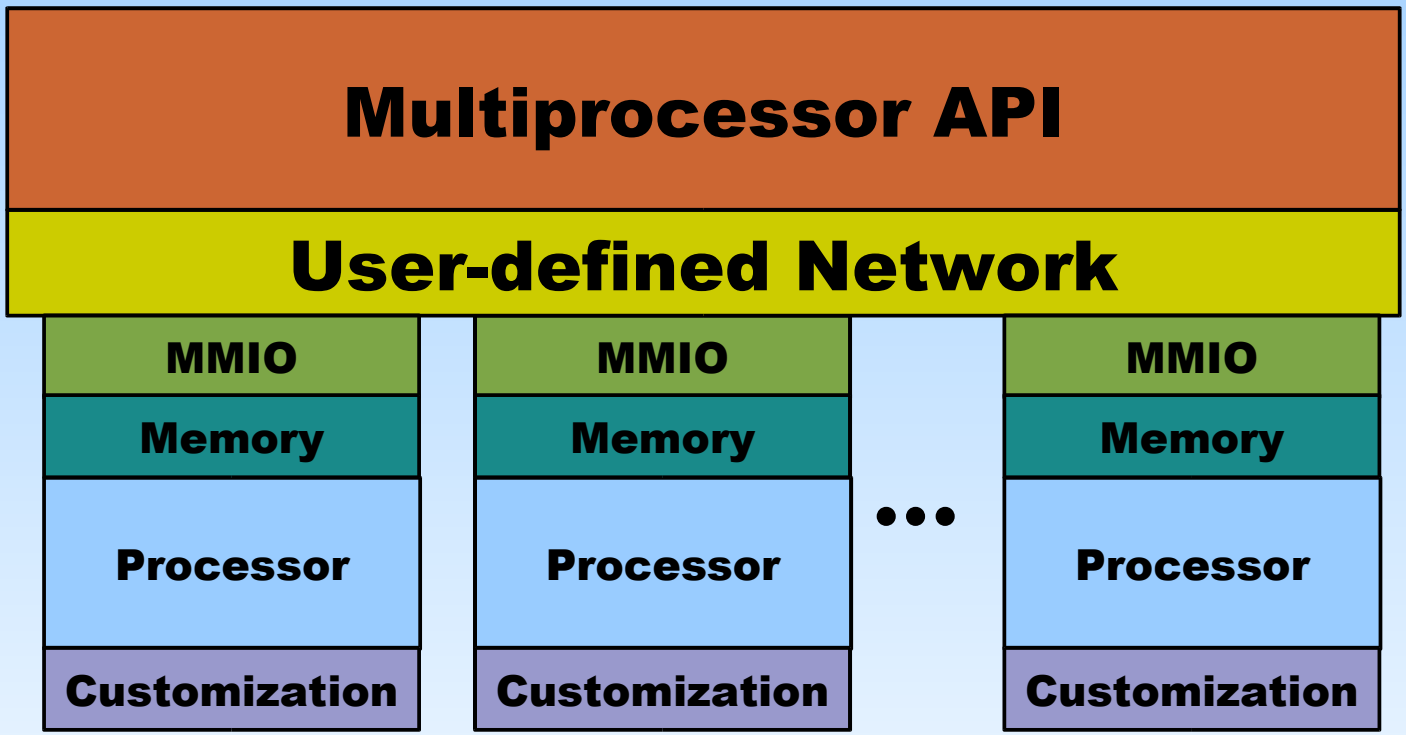
- Configurable Multiprocessing:
  - High performance
  - Power efficient
  - Very flexible
  - Highly programmable
  - Minimizes hardware design / verification effort
  - Leverages existing software IP

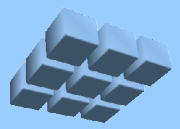


References

- Lance Hammond, Basem A. Nayfeh and Kunle Olkotun, A Single-Chip Multiprocessor, Computer, Volume 30, Number 9, IEEE Computer Society Press Los Alamitos, CA, USA, September 1997, Pages 79-85.
- "*The Future of Microprocessors*", David Patterson, U. California at Berkeley, June 2001. http://www.cs.berkeley.edu/~pattrsn/talks/NAE.ppt
- Chip Multiprocessing Resources: http://www.princeton.edu/~jdonald/research/cmp/
- Stanford Hydra Project: http://www-hydra.stanford.edu/
- Microprocessor Hall of Fame: http://www.intel.com/intel/intelis/museum/online/hist_micro/hof/tspecs.htm
- Microprocessor Quick Reference: http://www.intel.com/pressroom/kits/quickreffam.htm
- Arc International: http://www.arc.com/
- Sony / IBM / Toshiba "cell": http://www-3.ibm.com/chips/news/2001/0312_sony-toshiba.html

The Cmpware Multiprocessor Model





Extra Slides