Adding Low-Cost Hardware Barrier Support to Small Commodity Clusters

Torsten Höfler
Department of Computer Science
TU Chemnitz

June 24, 2006

Outline

Outline

Earth Simulator

- Global Barrier Counter (GBC)
- Flag registers within a processor node (Global Barrier Flag - GBF)

Earth Simulator Barrier

working principle:

1. Master node sets number of nodes into GBC
2. Control unit resets all GBFs of nodes
3. A completed node decrements GBC, and loops on GBF
4. When GBC=0 $\rightarrow$ control unit sets all GBFs
5. All nodes continue

$\Rightarrow$ constant barrier latency of 3.5μs between 2 and 512 nodes

BlueGene/L

- Independent Barrier Network
- Four independent Channels

BlueGene/L Barrier

working principle:

1. Global OR
2. Global AND by inverted logic
3. Signal is propagated to top of a binomial Tree and down
4. OR is used for Interrupts (halt machine)
5. AND is used for Barrier
6. Can be partitioned at specific borders

$\Rightarrow$ constant barrier latency of 1.5μs between 2 and 65536 nodes

Cray T3D

- Two Fetch&Increment Registers per Processor
- Global AND/OR barrier

Other Hardware Barriers

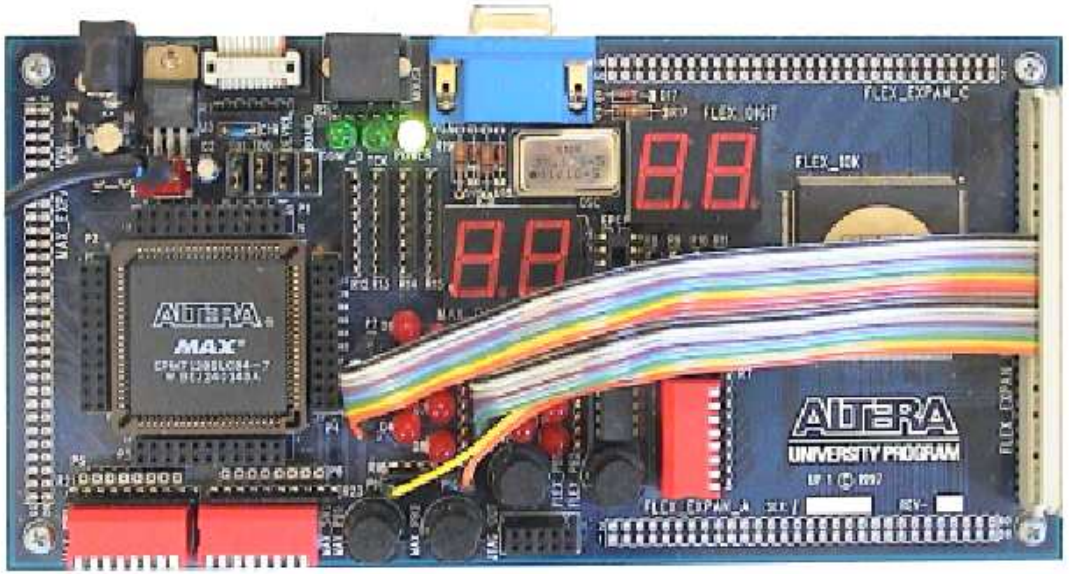
... many many more with same principles:

- Cray T3D
- Fujitsu VPP500
- Thinking Machines CM-5
- Purdue's Adapter
- ...

$\Rightarrow$ our approach is to support commodity clusters without changes in the machine itself

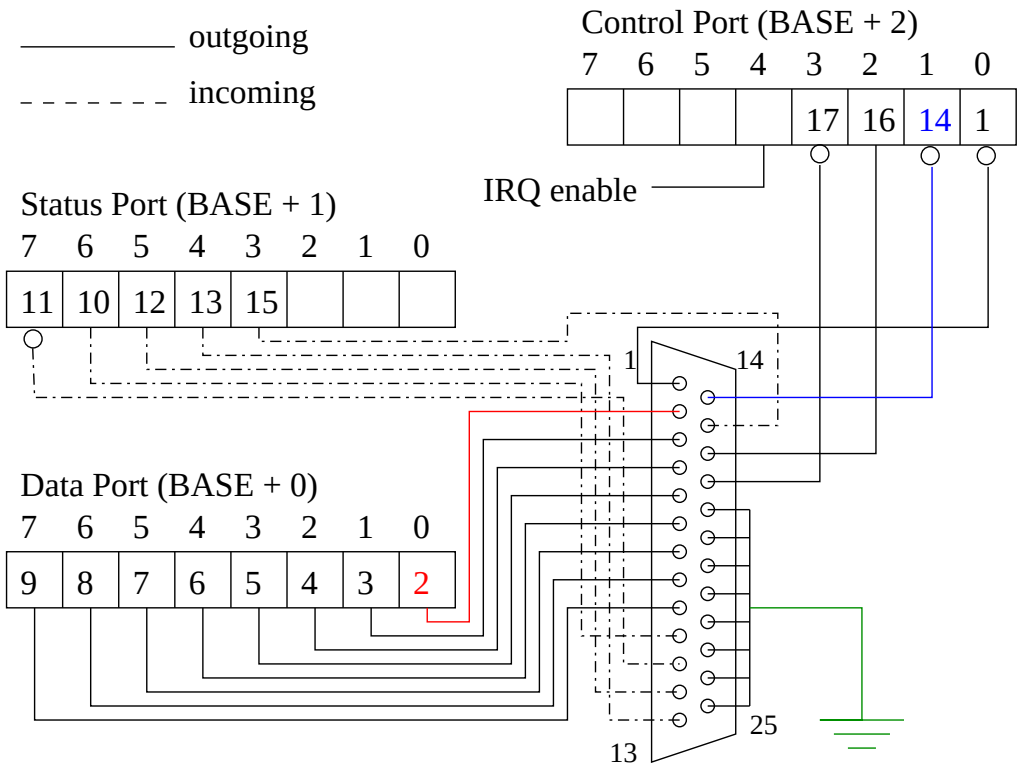
Outline

FPGA Based Prototype



- Simple and cheap design
- Prototype supports 1 barrier per node

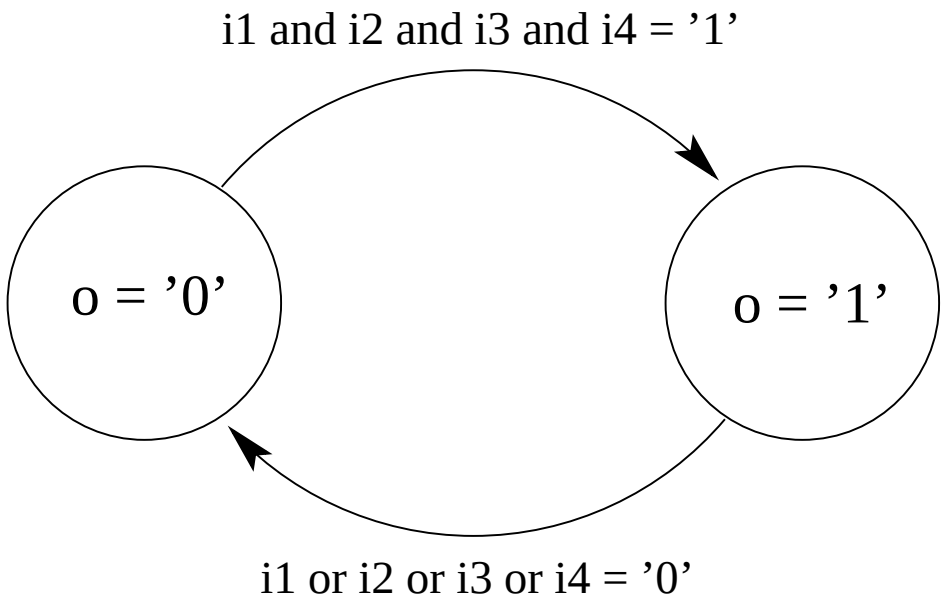
Parallel Port



- Three cables per node (`IN`, `OUT`, `GND`)
- Prototype supports 1 barrier per node

Outline

Two-state Machine



- Two states (2 FFs + $\lceil log_2 P \rceil$ 2-port ANDs/ORs)
- Very fast state transition
- `OUT` $\leftrightarrow$ `iP`, `IN` $\leftrightarrow$ `o`

Working Principle

Goal: minimize read/write Operations!

1. `init` only: read status (`IN`)
2. toggle status
3. write new status (`OUT`)
4. read status (`IN`) until toggled

$\rightarrow$ no ”packets”, constant Voltage-Level based

Scalability

Goal: Connect more than thousand nodes!

- Similar principle as for BlueGene/L
- AND/OR tree
- Propagating state up and down
- Two-state principle

Outline

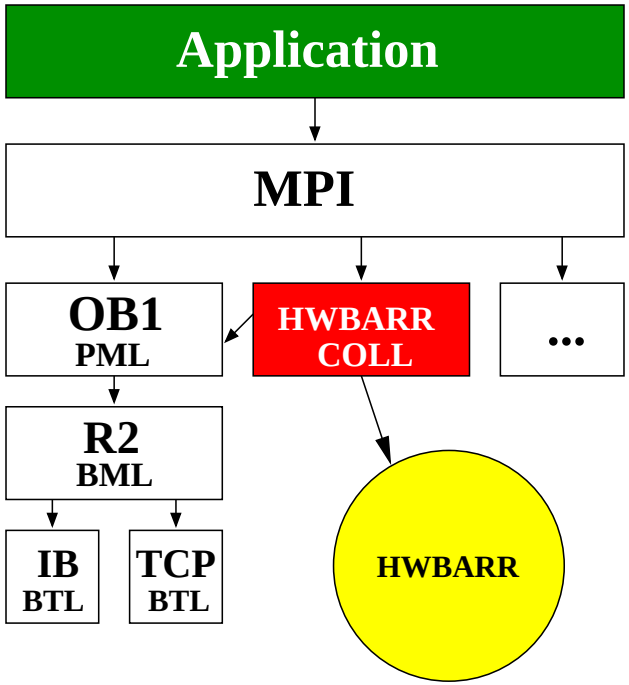
Accessing the Parallel Port

```
1  #define BASEPORT 0x378
   int main() {
     /* Set the data signals (D0–7) of the port to '0' */
     outb(0, BASEPORT);
     /* Read from the status port (BASE+1) */
6    printf("status: %d\n", inb(BASEPORT + 1));
   }
```

- Protoype uses `INB`, `OUTB`
- Requires root-access and OS adds overhead
- Kernel module with mmapped registers easily possible

Outline

Collective Module in Open MPI



- Implemented as collective Module in Open MPI
- Prototype supports only MPI_COMM_WORLD
- Requires to run as root

Outline

Performance Model

Variables:

1. t_b: Barrier latency
2. o_w: CPU overhead to write to the parallel port
3. o_r: CPU overhead to read from the parallel port
4. $o_p(P)$: Processing overhead of a state change
5. P: Number of processors

$\rightarrow$ toggle - write - read schema: $t_b = o_w + o_p(P) + o_r$

Parameter Benchmark

Benchmarked Parameters (4 2.4 GHz Xeon nodes):

- $o_w = 1.2\mu s$
- $o_r = 1.2\mu s$
- $o_p(P) = P \cdot 0.01\mu s$

$\rightarrow t_b = 1.2\mu s + 4 \cdot 0.01\mu s + 1.2\mu s = 2.44\mu s$

MPI Microbenchmark

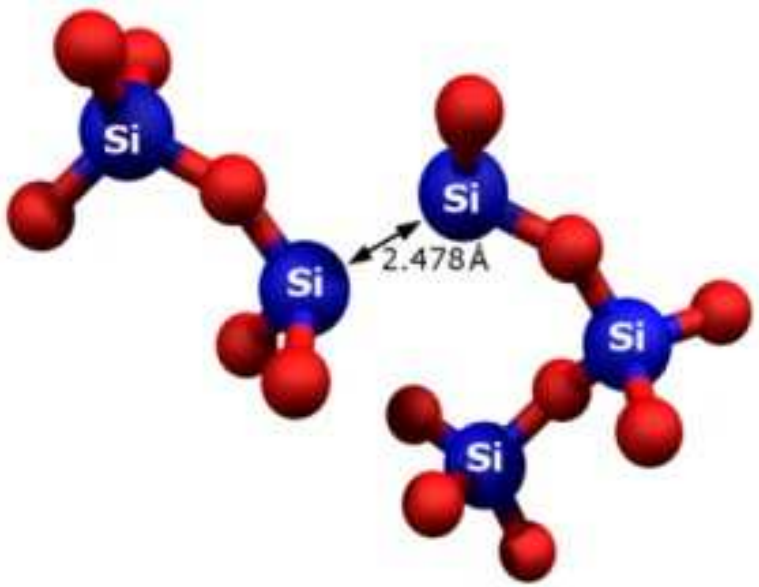
PMB-1 (4 2.4 GHz Xeon nodes):

- 1000 repetitions of MPI_BARRIER
- Average of 2.57μs
- Open MPI framework adds only 0.13μs
- cp. GigE, 4 nodes: $\approx 80\mu s$
- cp. IB, 4 nodes: $\approx 14\mu s$

$\longrightarrow$ compareable to commercial Hardware Barriers

Outline

Benchmarking Abinit



- Calculates electronic structures of solids
- Uses MPI_BARRIER for MPI_COMM_WORLD
- 8% MPI overhead
- 65% of MPI overhead is due to MPI_BARRIER

Abinit Results

Comparison between GigE and HWBARR:

- GigE: 4:34 min
- HWBARR: 4:27 min
- MPI overhead decreased by nearly 32%
- MPI_BARRIER overhead is halved

Conclusions

- Comparable to commercial hardware barriers
- Extensible design
- o_r/o_w can be reduced with memory mapping
- More wires per node could be used (5 in, 12 out)
- $\longrightarrow$ up to 2^{11} barriers
- $\longrightarrow$ incoming interrupt wire
- general OS support (e.g. /dev/barrier0)
- ...