

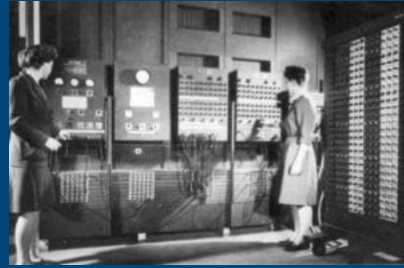
Today's Outline

- System history and evolution
 - Parallel Classification
 - System Architecture
 - Programming Model
 - Design Limitations
 - Future Technologies
 - FPGA
 - Cell Broadband Engine
 - GPU
-
-

In the Beginning - ENIAC

- Electronic Numerical Integrator and Computer
 - By most is considered to be the first electronic computer
 - Constructed by Penn's Moore School of Electrical Engineering from July, 1943
 - Unveiled at UPenn on February 15, 1946
 - At a delivered cost of \$500,000
-
-

ENIAC



- Weight: 27 tons
 - Components
 - 17,468 vacuum tubes
 - 7,200 crystal diodes
 - 1,500 relays
 - 70,000 resistors
 - 10,000 capacitors
 - 5 million hand-soldered joints
 - Roughly 8 feet by 3 feet by 100 feet
 - 150 kW of power
 - 5000 simple adds/subtracts per second
-
-

Other Computation Events in the 1940's

- 1944 - Relay-based Harvard-IBM MARK I provides vital calculations for the U.S. Navy
 - Grace Hopper becomes its programmer
 - 1945 - Computer 'bug' was termed by Grace Hopper when programming the MARK II
 - 1947 – Invention of the transistor
 - 1948 - IBM SSEC (Selective Sequence Electronic Calculator)
 - contains 12,000 tubes
-
-

Famous Last Words

- *“Where a calculator like the ENIAC today is equipped with 18,000 vacuum tubes and weighs 30 tons, computers in the future may have only 1,000 vacuum tubes and perhaps weigh only half a ton.”*
– *Popular Mechanics, March 1949*
-
-

1950's

- 1951 - First business computer, the Lyons Electronic Office (LEO)
 - 1951 - First commercial computer, the “First Ferranti MARK I” functional at Manchester University
 - 1951 - Unisys UNIVAC I
 - 1952 – First reliable magnetic drum memory
 - 1952 – IBM 701 introduced
-
-

More Famous Last Words

“I think there is a world market for about five computers”

-Thomas J. Watson Jr., chairman of IBM (1943)

1950's

- 1953 – IBM 701 sold to scientific community
 - 19 built and sold
 - 1954 – IBM 650 introduced
 - 1800 sold over its production lifetime
 - 1954 - FORTRAN
 - 1955 – IBM 702
-
-

1950's

- 1955 – Bell Labs introduces first all transistor computer
 - 1955 – ENIAC shut down for final time
 - 1957 – IBM announces no more vacuum tube computers, releases first all transistor computer (contains 2000 transistors)
 - The first microchip was demonstrated on September 12, 1958.
-
-

Let's jump to 1969

- AT&T Bell Labs develops UNIX
 - AMD founded
 - First laser printer (Xerox)
 - Advanced Research Projects Agency Network, ARPANET
 - CDC 7600, first supercomputer
-
-

CDC 7600

- small-core memory of 64k 60-bit words
- clock speed of 27 nanoseconds
- Instruction pipeline
- Peripheral processors
- No software!
- Prone to breakdown!!!



Cray 1 - 1976

- Hand Crafted!
- Hundreds of circuit boards and thousands of wires that had to fit just right
- Special cooling required
- Needed room for:
 - the big main unit,
 - the huge power supply next door,
 - couple of mainframe's just to feed data
- Seymour Cray's secret - *“Figure out how to build it as fast as possible, completely disregarding the cost of construction.”*



Cray 1 - Specs



- "C" shape
 - enabled integrated circuits to be closer together
 - No wire in the system > four feet long
 - low-density/very high-speed ECL circuits
 - required special cooling
 - 133 megaflops
 - 8 MB memory
 - First installed at LLNL
 - 85 built
 - \$5 - \$8.8 million
-

Cray 1 - Specs



- 200,000 specialized low-density ECL integrated circuits
 - Programming Model:
 - Vector processing
 - 8 vector registers, 64 64-bit words each
 - Software:
 - Cray Operating System (COS),
 - Cray Fortran Compiler,
 - Cray Assembler Language
-

Vector Processing

- Operate on all of the data "from here to here" to all of the data "from there to there"
 - Reads a single instruction from memory, and "knows" that the next address will be one larger than the last
 - Significant performance improvement
-
-

Approaching the 'modern' era

- 'On chip' parallelism
 - multiple registers
 - instruction and data pipelines
 - vector processing
 - low level instructions, compilers that understand them, programmer who know how to use them
 - Vector processing easy
 - Expensive, niche market machines
-
-

Flynn's Taxonomy

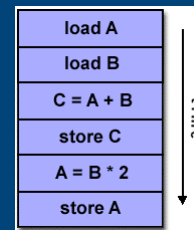
- One of the more widely used classifications, in use since 1966
 - Distinguishes multi-processor computer architectures according to how they can be classified along the two independent dimensions
 - Instruction
 - Data
 - Each of these dimensions can have only one of two possible states
 - single
 - multiple
-
-

Flynn's Taxonomy

- | | |
|--------|------|
| • SISD | SIMD |
| • MISD | MIMD |
-
-

Single Instruction, Single Data

- Single instruction: only one instruction stream is being acted on by the CPU during any one clock cycle
- Single data: only one data stream is being used as input during any one clock cycle
- Deterministic execution
- This is the oldest and until recently, the most prevalent form of computer



Single Instruction, Multiple Data

- Single instruction: All processing units execute the same instruction at any given clock cycle
- Multiple data: Each processing unit can operate on a different data element
- This type of machine typically has an instruction dispatcher, a very high-bandwidth internal network, and a very large array of very small-capacity instruction units
- Best suited for specialized problems characterized by a high degree of regularity, such as image processing
- Synchronous (lockstep) and deterministic execution

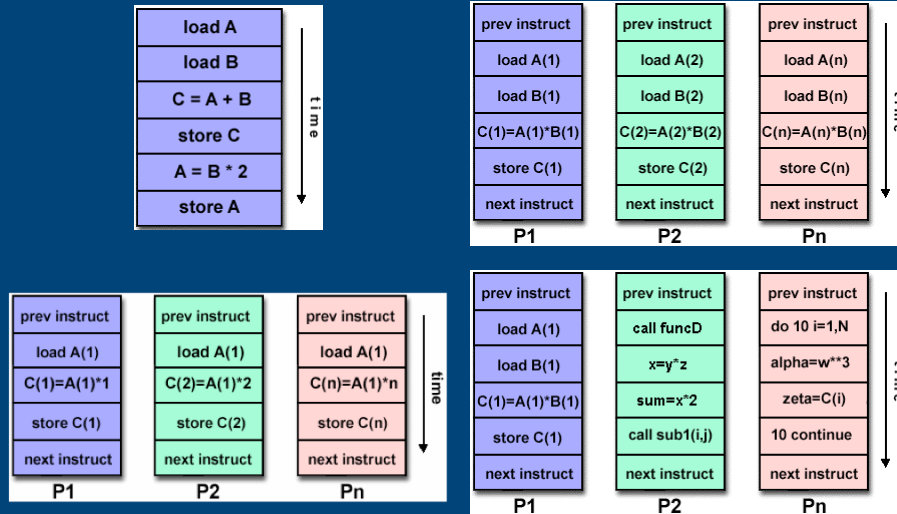
Mult. Instruction, Single data

- Single data stream fed to multiple processing units
 - Each processing unit operates on the data independently via independent instruction streams
 - Few actual examples of this class of parallel computer have ever existed
 - Some conceivable uses might be:
 - multiple frequency filters operating on a single signal stream
 - multiple cryptography algorithms attempting to crack a single coded message.
-
-

Mult. Instruction, Mult. Data

- Most modern computers fall into this category
 - Multiple Instruction: every processor may be executing a different instruction stream
 - Multiple Data: every processor may be working with a different data stream
 - Execution can be synchronous or asynchronous, deterministic or non-deterministic
 - Examples: most current supercomputers, networked parallel computer, 'grid' computing, SMP, some PC's
-
-

A graphical Flynn



Early Parallel Systems

- Thinking Machines CM1
 - hypercube arrangement
 - 1000's of very simple processors
 - each with its own RAM
 - SIMD
- CM2
 - Up to 64k single-bit processors
 - 1 fp coprocessor per 32 procs
 - hypercube
- CM5
 - MIMD
 - Fat tree
 - SPARC, SuperSPARC
 - as seen in Jurassic Park

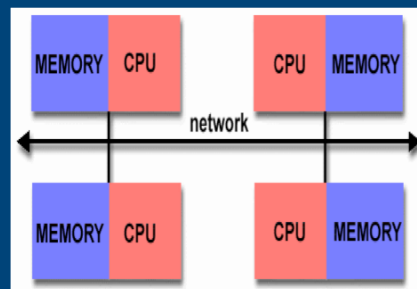
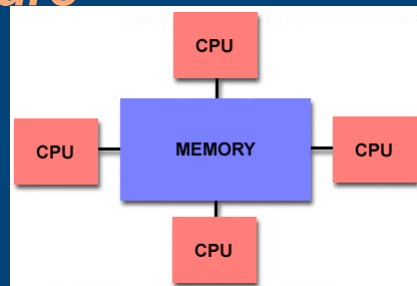


Next Generation

- IBM SP
- SGI Origin
- NEC SX series
- COTS
- Model being driven by cost!

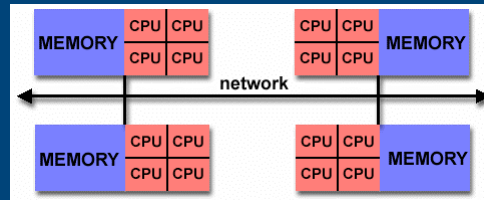
Memory Architecture

- Shared Memory
 - UMA
 - NUMA
- Distributed Memory



Memory Architecture

- Hybrid



Shared Memory

- Advantages
 - Global address space provides a user-friendly programming perspective to memory
 - Data sharing between tasks is both fast and uniform due to the proximity of memory to CPUs
- Disadvantages
 - Primary disadvantage is the lack of scalability between memory and CPUs
 - Adding more CPUs can geometrically increase traffic on the shared memory-CPU path
 - increase traffic associated with cache/memory management
 - Expensive

Distributed Memory

- Advantages
 - Memory is scalable with number of processors
 - Each processor can rapidly access its own memory without interference and without the overhead incurred with trying to maintain cache coherency
 - Cost effectiveness: can use commodity, off-the-shelf processors and networking.
 - Disadvantages
 - Programmer responsible for data communication details
 - It may be difficult to map existing data structures, based on global memory, to this memory organization
 - NUMA access times
-
-

Hybrid Model

- Shared memory component is usually a cache coherent SMP machine
 - Processors on a given SMP can address that machine's memory as global.
 - Distributed memory component is the networking of multiple SMPs
 - SMPs know only about their own memory
 - Network communications are required to move data from one SMP to another.
 - Most current trend
 - Advantages and Disadvantages: whatever is common to both shared and distributed memory architectures
-
-

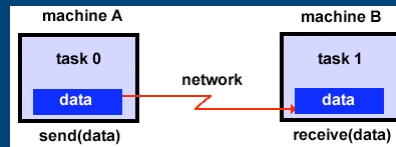
Programming Models

- Common models
 - Shared Memory
 - Threads
 - Message passing
 - Data Parallel
 - Hybrid
 - Parallel programming models exist as an abstraction above hardware and memory architectures
 - Although it might not seem apparent, these models are NOT specific to a particular type of machine or memory architecture
 - These models can theoretically be implemented on any underlying hardware
-
-

Shared Memory Model

- All tasks share a common address space
 - read and write asynchronously
 - Various mechanisms such as locks / semaphores may be used to control access to the shared memory
 - No notion of data "ownership"
 - no need to specify explicitly the communication of data between tasks
 - Program development can often be simplified
 - A disadvantage in terms of performance is that it becomes more difficult to understand and manage data locality
-
-

Distributed Memory Model



- Message Passing
 - de facto standard today
 - Programmer is responsible for determining all parallelism
 - Set of tasks that use their own local memory during computation
 - Multiple tasks can reside on the same physical machine as well across an arbitrary number of machines.
 - Tasks exchange data through communications by sending and receiving messages
 - Data transfer often needs cooperative operations to be performed by each process.
-
-

Hybrid Model

- Some combination of above
 - Shared memory / MPI most common
 - Threads / MPI, etc.
 - Lends itself well to the increasingly common hardware environment of networked SMP machines
 - Idea is to leverage benefits from each model
-
-

Discussion

- Advantages / Disadvantages?
-
-

Future Direction

- FPGA
 - Cell Broadband Engine
 - GPU
-
-

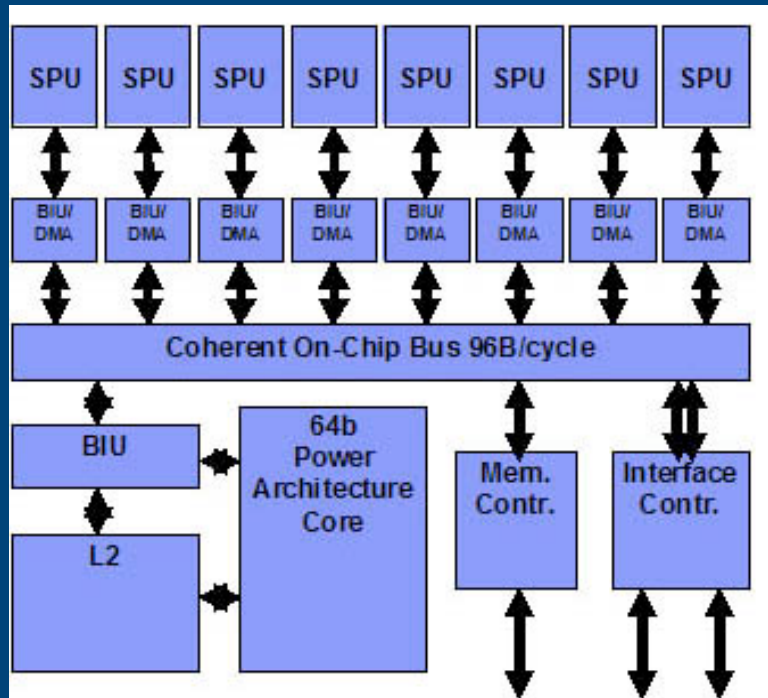
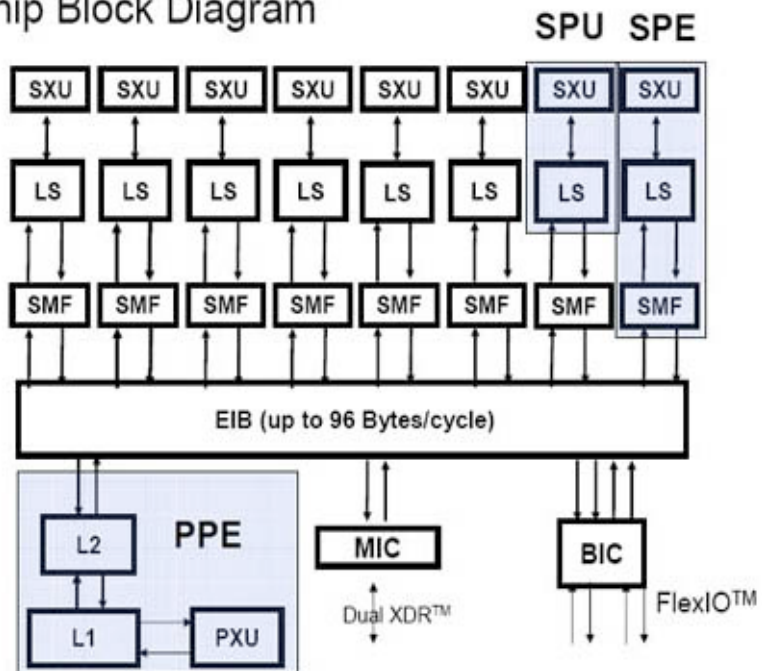
FPGA

- Field programmable gate array
 - A semiconductor device containing programmable logic components and programmable interconnects
 - Can be reprogrammed at "run time"
 - reconfigurable computing or reconfigurable systems
 - Current FPGA tools, however, do not fully support this methodology
 - Generally slower and hotter than ASICs
 - Application area is any algorithm that can make use of the massive parallelism offered by their architecture
-
-

Cell Broadband Engine

- Jointly developed by STI, an alliance of Sony, Toshiba, and IBM
 - combines a light-weight general-purpose processor with multiple GPU-like coprocessors into a coordinated whole, a feat which involves a novel memory coherence architecture for which IBM received many patents
 - scientific calculations 3 to 12 times faster than any desktop processor at a similar clock speed
 - Software adoption remains a key issue in whether Cell ultimately delivers on its performance potential
-
-

Cell Chip Block Diagram



First Application?

- Roadrunner - 1st petaflop machine
 - Opteron / Cell hybrid
 - Cell as accelerator
 - 16k processors
 - 1:1 match
 - Los Alamos National Lab
 - 2008 deployment
-
-

GPU Accelerator

- Uses high end graphics card attached via PCI-E bus
 - *PeakStream Computing*
 - C/C++ API
 - Fortran coming
 - Single Precision! (but DP coming soon)
-
-

***As always, when trying to predict
the future...***

“640 K [of computer memory] ought to be enough
for anybody.”

- *Bill Gates, 1981*

“This telephone has too many shortcomings
to be seriously considered as a means of
communication. The device is inherently of
no value to us.”

-*Western Union internal memo (1876)*

Credits

- Blaise Barney "Introduction to Parallel Computing"
http://www.llnl.gov/computing/tutorials/parallel_comp/
 - Wikipedia
 - ENIAC - <http://en.wikipedia.org/wiki/ENIAC>
 - CDC 7600 -
 - Connection Machines - http://en.wikipedia.org/wiki/Connection_Machine
 - FPGA - <http://en.wikipedia.org/wiki/FPGA>
 - PeakStream Computing
 - IBM Cell Broadband Engine -
 - National Center for Atmospheric Research
-
-