Instruction Set Architecture

- Class of ISA
  - General-purpose registers
  - Register-memory vs load-store
- RISC-V registers
  - 32 g.p., 32 f.p.

Register	Name	Use	Saver
x0	zero	constant 0	n/a
x1	ra	return addr	caller
x2	sp	stack ptr	callee
x3	gp	gbl ptr	
x4	tp	thread ptr	
x5-x7	t0-t2	temporaries	caller
x8	s0/fp	saved/ frame ptr	callee
x9	s1	saved	callee
x10-x17	a0-a7	arguments	caller
x18-x27	s2-s11	saved	callee
x28-x31	t3-t6	temporaries	caller
f0-f7	ft0-ft7	FP temps	caller
f8-f9	fs0-fs1	FP saved	callee
f10-f17	fa0-fa7	FP arguments	callee
f18-f27	fs2-fs21	FP saved	callee
f28-f31	ft8-ft11	FP temps	caller

Instruction Set Architecture

- Memory addressing
  - RISC-V: byte addressed, aligned accesses faster
- Addressing modes
  - RISC-V: Register, immediate, displacement (base+offset)
  - Other examples: autoincrement, indexed, PC-relative
- Types and size of operands
  - RISC-V: 8-bit, 32-bit, 64-bit

Instruction Set Architecture

- Operations
  - RISC-V: data transfer, arithmetic, logical, control, floating point
  - See Fig. 1.5 in text
- Control flow instructions
  - Use content of registers (RISC-V) vs. status bits (x86, ARMv7, ARMv8)
  - Return address in register (RISC-V, ARMv7, ARMv8) vs. on stack (x86)
- Encoding
  - Fixed (RISC-V, ARMv7/v8 except compact instruction set) vs. variable length (x86)

Trends in Technology

- The architect must take technology changes into consideration.
- Technology changes very rapidly, architecture changes slowly (we are still using x86 compatible systems).
- Usually, the architecture you are designing now, will be used with “next technology”.

Trends in Technology

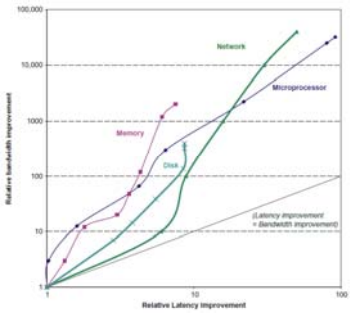
- Integrated circuit technology (Moore’s Law)
  - Transistor density: 35%/year – Not any more (The end of Moore’s lay).
  - Die size: 10-20%/year
  - Integration overall: 40-55%/year
- DRAM capacity: 25-40%/year (slowing)
  - 8 Gb (2014), 16 Gb (2019), possibly no 32 Gb
- Flash capacity: 50-60%/year
  - Roughly doubling every 2 years
  - 8-10X cheaper/bit than DRAM
- Magnetic disk capacity: recently slowed to 5%/year
  - Density increases may no longer be possible, maybe increase from 7 to 9 platters
  - 8-10X cheaper/bit then Flash
  - 200-300X cheaper/bit than DRAM

Bandwidth and Latency

- Bandwidth or throughput
  - Total work done in a given time
  - 32,000-40,000X improvement for processors
  - 300-1200X improvement for memory and disks
- Latency or response time
  - Time between start and completion of an event
  - 50-90X improvement for processors
  - 6-8X improvement for memory and disks

Bandwidth and Latency

Trends in Technology



Log-log plot of bandwidth and latency milestones

 21

Transistors and Wires

Trends in Technology

- Feature size
  - Minimum size of transistor or wire in x or y dimension
  - 10 microns in 1971 to .011 microns in 2017
  - Transistor performance scales linearly
    - Wire delay does not improve with feature size!
  - Integration density scales quadratically

 22

Power and Energy

Trends in Power and Energy

- Problem: Get power in, get power out
- Thermal Design Power (TDP)
  - Characterizes sustained power consumption
  - Used as target for power supply and cooling system
  - Lower than peak power (1.5X higher), higher than average power consumption
- Clock rate can be reduced dynamically to limit power consumption
- Energy per task is often a better measurement

 23

Dynamic Energy and Power

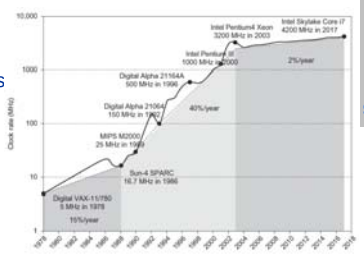
- Dynamic energy
  - Transistor switch from 0 -> 1 or 1 -> 0
  - ½ x Capacitive load x Voltage^2
- Dynamic power
  - ½ x Capacitive load x Voltage^2 x Frequency switched
- Reducing clock rate reduces power, not energy

Dynamic Energy and Power

- Example:

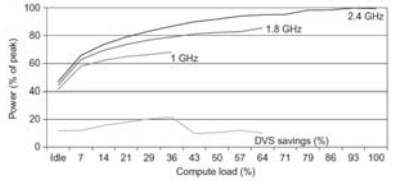
Power

- Intel 80386 consumed ~ 2 W
- 3.3 GHz Intel Core i7 consumes 130 W
- Heat must be dissipated from 1.5 x 1.5 cm chip
- This is the limit of what can be cooled by air



Reducing Power

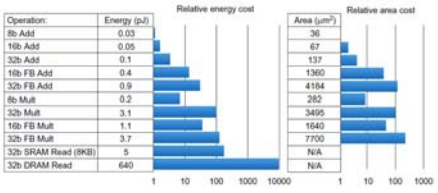
- Techniques for reducing power:
  - Do nothing well
  - Dynamic Voltage-Frequency Scaling



  - Low power state for DRAM, disks
  - Overclocking, turning off cores

Static Power

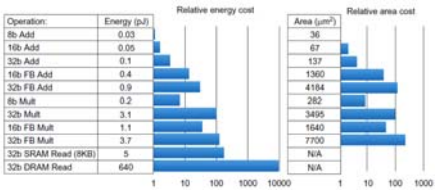
- Static power consumption
  - 25-50% of total power
  - $Current_{static}$ x Voltage
  - Scales with number of transistors
  - To reduce: power gating



Operation:	Energy (pJ)	Area (μm²)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FB Add	0.4	1360
32b FB Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FB Mult	1.1	1640
32b FB Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A

Shift in Architecture

- Dark silicon: more silicon (devices) that could be powered at the same time.
- What to include in your design?
- Domain Specific Architecture: Do one thing really efficient



Operation:	Energy (pJ)	Area (μm²)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FB Add	0.4	1360
32b FB Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FB Mult	1.1	1640
32b FB Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A

Trends in Cost

- Cost driven down by learning curve
  - Yield
- DRAM: price closely tracks cost
- Microprocessors: price depends on volume
  - 10% less for each doubling of volume
- Effect of "Commoditization"
  - Multiple vendors selling "*essentially*" the same product in large volume (low end computers).
  - Many suppliers compete for components
- Operational Expenses (WSC)

 30

Integrated Circuit Cost

- Integrated circuit

$$Cost\ of\ integrated\ circuit = \frac{Cost\ of\ die + Cost\ of\ testing\ die + Cost\ of\ packaging\ and\ final\ test}{Final\ test\ yield}$$

$$Cost\ of\ die = \frac{Cost\ of\ wafer}{Dies\ per\ wafer \times Die\ yield}$$

$$Dies\ per\ wafer = \frac{\pi \times (Wafer\ diameter/2)^2}{Die\ area} - \frac{\pi \times Wafer\ diameter}{\sqrt{2} \times Die\ area}$$

- Bose-Einstein formula:

$$Die\ yield = Wafer\ yield \times 1/(1 + Defects\ per\ unit\ area \times Die\ area)^N$$

- Defects per unit area = 0.016-0.057 defects per square cm (2010)
- N = process-complexity factor = 11.5-15.5 (40 nm, 2010)

 31

Dependability

- SLA guarantees a certain level of dependability/availability
- Module reliability
  - Mean time to failure (MTTF)
  - Mean time to repair (MTTR)
  - Mean time between failures (MTBF) = MTTF + MTTR
  - Availability = MTTF / MBTF
- Failure rate (FIT)=10^9/MTTF (failure per billion hours of operation)

 32

Cost of Unavailability -- Servers

Application	Cost of downtime per hour	Annual losses with downtime of 1% (87.6 h/year)	0.5% (43.8 h/year)	0.1% (8.8 h/year)
Brokerage service	$4,000,000	$350,400,000	$175,200,000	$35,000,000
Energy	$1,750,000	$153,300,000	$76,700,000	$15,300,000
Telecom	$1,250,000	$109,500,000	$54,800,000	$11,000,000
Manufacturing	$1,000,000	$87,600,000	$43,800,000	$8,800,000
Retail	$650,000	$56,900,000	$28,500,000	$5,700,000
Health care	$400,000	$35,000,000	$17,500,000	$3,500,000
Media	$50,000	$4,400,000	$2,200,000	$400,000

Figure 1.3 Costs rounded to nearest $100,000 of an unavailable system are shown by analyzing the cost of downtime (in terms of immediately lost revenue), assuming three different levels of availability, and that downtime is distributed uniformly. These data are from Landstrom (2014) and were collected and analyzed by Contingency Planning Research.

 33

Dependability

- Example
  - 10 disks 1,000,000-hour MTTF
  - 1 ATA controller 500,000-hour MTTF
  - 1 Power supply 200,000-hour MTTF
  - 1 Fan 200,000-hour MTTF
  - 1 ATA cable 1,000,000-hour MTTF
- Assume lifetimes are exponentially distributed and failures are independent
- Calculate MTTF

 34

Redundancy

- Consider the previous example.
- 200,000 MTTF for power supplies
- What is the effect of adding one more power supply (system fails if both power supplies failed at the same time).

 35

Measuring Performance

- Typical performance metrics:
  - Response time
  - Throughput
- Speedup of X relative to Y
  - Execution time$_Y$ / Execution time$_X$
- Execution time
  - Wall clock time: includes all system overheads
  - CPU time: only computation time
- Benchmarks
  - Kernels (e.g. matrix multiply)
  - Toy programs (e.g. sorting)
  - Synthetic benchmarks (e.g. Dhrystone)
  - Benchmark suites (e.g. SPEC06fp, TPC-C)

 36

Reporting

- Many programs, how can we capture performance using a single number?

	P1	P2	P3
Machine-A	10	8	25
Machine-B	12	9	20
Machine-C	8	8	30

- Sum of execution time
- Sum of weighted execution time
- Geometric mean of execution time

 37

Measuring Performance

- Many programs, how can we capture performance using a single number?

	P1	P2	P3
Machine-A	10	8	25
Machine-B	12	9	20
Machine-C	8	8	30

- Sum of execution time
- Sum of weighted execution time
- Geometric mean of execution time

 38

Measuring Performance

- Many programs, how can we capture performance using a single number?

	P1	P2	P3
Machine-A	10	8	25
Machine-B	12	9	20
Machine-C	8	8	30

- Geometric mean of execution time

Measuring Performance

- Time = TC × CPI × IC
- Must be reproducible
- Complete description of the computer and compiler flags.
- Usually, compared to a standard machine execution time SPECRatioA = T_{ref}/T_A.
- Geometric mean

Measuring Performance

Benchmarks	Sun Ultra Enterprise 2 time (seconds)	AMD A10-6800K time (seconds)	SPEC 2006Cint ratio	Intel Xeon E5-2690 time (seconds)	SPEC 2006Cint ratio	AMD/Intel times (seconds)	Intel/AMD SPEC ratios
perlbench	9770	401	24.36	261	37.43	1.54	1.54
bzip2	9650	505	19.11	422	22.87	1.20	1.20
gcc	8050	490	16.43	227	35.46	2.16	2.16
mcf	9120	249	36.63	153	59.61	1.63	1.63
gobmk	10,490	418	25.10	382	27.46	1.09	1.09
hmmer	9330	182	51.26	120	77.75	1.52	1.52
sjeng	12,100	517	23.40	383	31.59	1.35	1.35
libquantum	20,720	84	246.08	3	7295.77	29.65	29.65
h264ref	22,130	611	36.22	425	52.07	1.44	1.44
omnetpp	6250	313	19.97	153	40.85	2.05	2.05
astar	7020	303	23.17	209	33.59	1.45	1.45
xalancbmk	6900	215	32.09	98	70.41	2.19	2.19
Geometric mean			31.91		63.72	2.00	2.00

Principles of Computer Design

Principles

- Take Advantage of Parallelism
  - e.g. multiple processors, disks, memory banks, pipelining, multiple functional units
- Principle of Locality
  - Reuse of data and instructions
- Focus on the Common Case
  - Amdahl's Law

$$Execution\ time_{new} = Execution\ time_{old} \times \left((1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}\right)$$

$$Speedup_{overall} = \frac{1}{(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

 42

Principles of Computer Design

Principles

- The Processor Performance Equation

$$CPU\ time = CPU\ clock\ cycles\ for\ a\ program \times Clock\ cycle\ time$$

$$CPU\ time = \frac{CPU\ clock\ cycles\ for\ a\ program}{Clock\ rate}$$

$$CPI = \frac{CPU\ clock\ cycles\ for\ a\ program}{Instruction\ count}$$

$$CPU\ time = Instruction\ count \times Cycles\ per\ instruction \times Clock\ cycle\ time$$

$$\frac{Instructions}{Program} \times \frac{Clock\ cycles}{Instruction} \times \frac{Seconds}{Clock\ cycle} = \frac{Seconds}{Program} = CPU\ time$$

 43

Principles of Computer Design

Principles

- Different instruction types having different CPIs

$$CPU\ clock\ cycles = \sum_{i=1}^{n} IC_i \times CPI_i$$

$$CPU\ time = \left(\sum_{i=1}^{n} IC_i \times CPI_i\right) \times Clock\ cycle\ time$$

 44

Principles of Computer Design

- Different instruction types having different CPIs

$$\text{CPU clock cycles} = \sum_{i=1}^{n} IC_i \times CPI_i$$

$$\text{CPU time} = \left(\sum_{i=1}^{n} IC_i \times CPI_i \right) \times \text{Clock cycle time}$$

Fallacies and Pitfalls

- All exponential laws must come to an end
  - Dennard scaling (constant power density)
    - Stopped by threshold voltage
  - Disk capacity
    - 30-100% per year to 5% per year
  - Moore's Law
    - Most visible with DRAM capacity
    - ITRS disbanded
    - Only four foundries left producing state-of-the-art logic chips
    - 11 nm, 3 nm might be the limit

Fallacies and Pitfalls

- Microprocessors are a silver bullet
  - Performance is now a programmer's burden
- Falling prey to Amdahl's Law
- A single point of failure
- Hardware enhancements that increase performance also improve energy efficiency, or are at worst energy neutral
- Benchmarks remain valid indefinitely
  - Compiler optimizations target benchmarks

Fallacies and Pitfalls

- The rated mean time to failure of disks is 1,200,000 hours or almost 140 years, so disks practically never fail
  - MTTF value from manufacturers assume regular replacement
- Peak performance tracks observed performance
- Fault detection can lower availability
  - Not all operations are needed for correct execution