Recall: Limits to Multi-Issue Machines

- Multi-issue: simple matter of accounting
  - Must do dataflow analysis across multiple instructions simultaneously
  - Rename table updated as if instructions happened serially!

- To sustain: need execution bandwidth+commit bandwidth
  - To sustain ILP of X need at least
    - X-way issue, > X execution bandwidth (for mix), X way commit

- Inherent limitations of ILP
  - 1 branch in 5: How to keep a 5-way superscalar busy?
  - Latencies of units: many operations must be scheduled
  - Need about Pipeline Depth x No. Functional Units of independent instructions to keep fully busy
  - Increase ports to Register File
    - VLIW example needs 7 read and 3 write for Int. Reg. & 5 read and 3 write for FP reg
  - Increase ports to memory
  - Current state of the art: Many hardware structures (such as issue/rename logic) has delay proportional to square of number of instructions issued/cycle

Recall: Upper Limit to ILP: Ideal Machine

Recall: More Realistic HW: Branch Impact
The Five Classic Components of a Computer

- Processor
- Control
- Datapath
- Memory
- Output

Today’s Topics:
- Recap last lecture
- Locality and Memory Hierarchy
- Administrivia
- SRAM Memory Technology
- DRAM Memory Technology
- Memory Organization

Recall: Memory Hierarchy of a Modern Computer System

- By taking advantage of the principle of locality:
  - Present the user with as much memory as is available in the cheapest technology.
  - Provide access at the speed offered by the fastest technology.

Recall: Who Cares About the Memory Hierarchy?

- Processor-DRAM Memory Gap (latency)
  - “Moore’s Law”
  - “Less’ Law?”

Memory Hierarchy Technology

- Random Access:
  - “Random” is good: access time is the same for all locations
  - DRAM: Dynamic Random Access Memory
    - High density, low power, cheap, slow
    - Dynamic: need to be “refreshed” regularly
  - SRAM: Static Random Access Memory
    - Low density, high power, expensive, fast
    - Static: content will last “forever” (until lose power)

- “Non-so-random” Access Technology:
  - Access time varies from location to location and from time to time
    - Examples: Disk, CDROM, DRAM page-mode access

- Sequential Access Technology: access time linear in location (e.g., Tape)
  - The next two lectures will concentrate on random access technology
    - The Main Memory: DRAMs + Caches: SRAMs
Main Memory Background

- **Performance of Main Memory:**
  - **Latency:** Cache Miss Penalty
    - **Access Time:** time between request and word arrives
    - **Cycle Time:** time between requests
  - **Bandwidth:** I/O & Large Block Miss Penalty (L2)

- **Main Memory is DRAM:** Dynamic Random Access Memory
  - Dynamic since needs to be refreshed periodically (8 ms)
  - Addresses divided into 2 halves (Memory as a 2D matrix):
    - RAS or Row Access Strobe
    - CAS or Column Access Strobe

- **Cache uses SRAM:** Static Random Access Memory
  - No refresh (6 transistors/bit vs. 1 transistor)
  - **Size:** DRAM/SRAM - 4-8
  - **Cost/Cycle time:** SRAM/DRAM - 8-16

Random Access Memory (RAM) Technology

- **Why do computer designers need to know about RAM technology?**
  - Processor performance is usually limited by memory bandwidth
  - As IC densities increase, lots of memory will fit on processor chip
    - Instruction cache
    - Data cache
    - Write buffer

- **What makes RAM different from a bunch of flip-flops?**
  - **Density:** RAM is much denser

Static RAM Cell

- 6-Transistor SRAM Cell
- **Write:**
  1. Drive bit lines (bit=1, bit=0)
  2. Select row
- **Read:**
  1. Precharge bit and bit to Vdd or Vdd/2 => make sure equal!
  2. Select row
  3. Cell pulls one line low
  4. Sense amp on column detects difference between bit and bit

Typical SRAM Organization: 16-word x 4-bit

Q: Which is longer: word line or bit line?
Write Enable is usually active low (WE_L)

Din and Dout are combined to save pins:
- A new control signal, output enable (OE_L) is needed
- WE_L is asserted (Low), OE_L is disasserted (High)
  - D serves as the data input pin
- WE_L is disasserted (High), OE_L is asserted (Low)
  - D is the data output pin
- Both WE_L and OE_L are asserted:
  - Result is unknown. Don’t do that!!!

Although could change VHDL to do what desire, must do the best with what you’ve got (vs. what you need)

Problems with SRAM

- Six transistors use up a lot of area
- Consider a “Zero” is stored in the cell:
  - Transistor N1 will try to pull “bit” to 0
  - Transistor P2 will try to pull “bit bar” to 1
- But bit lines are precharged to high: Are P1 and P2 necessary?

Main Memory Deep Background

- “Out-of-Core”, “In-Core,” “Core Dump”?
- “Core memory”?
- Non-volatile, magnetic
- Lost to 4 Kbit DRAM (today using 64Mbit DRAM)
- Access time 750 ns, cycle time 1500-3000 ns
1-Transistor Memory Cell (DRAM)

° Write:
  • 1. Drive bit line
  • 2. Select row

° Read:
  • 1. Precharge bit line to Vdd/2
  • 2. Select row
  • 3. Cell and bit line share charges
    - Very small voltage changes on the bit line
  • 4. Sense (fancy sense amp)
    - Can detect changes of ~1 million electrons
  • 5. Write: restore the value

° Refresh
  • 1. Just do a dummy read to every cell.

DRAM Capacitors: more capacitance in a small area

° Trench capacitors:
  • Logic ABOVE capacitor
  • Gain in surface area of capacitor
  • Better Scaling properties
  • Better Planarization

° Stacked capacitors
  • Logic BELOW capacitor
  • Gain in surface area of capacitor
  • 2-dim cross-section quite small

Classical DRAM Organization (square)

° Row and Column Address together:
  • Select 1 bit a time

DRAM logical organization (4 Mbit)

° Square root of bits per RAS/CAS
**DRAM physical organization (4 Mbit)**

- **Column Address**
- **I/O**
- **I/O**
- **I/O**
- **Row Address**
- **I/O**
- **I/O**
- **I/O**
- **I/O**
- **I/O**
- **I/O**
- **I/O**
- **I/O**

**Block 0**  
**Block 3**

---

**Logic Diagram of a Typical (Asynchronous) DRAM**

° Control Signals (RAS_L, CAS_L, WE_L, OE_L) are all active low
° Din and Dout are combined (D):
  - WE_L is asserted (Low), OE_L is disasserted (High)
    - D serves as the data input pin
  - WE_L is disasserted (High), OE_L is asserted (Low)
    - D is the data output pin
° Row and column addresses share the same pins (A)
  - RAS_L goes low: Pins A are latched in as row address
  - CAS_L goes low: Pins A are latched in as column address
  - RAS/CAS edge-sensitive

---

**DRAM Read Timing**
° Every DRAM access begins at:
  - The assertion of the RAS_L
  - 2 ways to read: early or late v. CAS

**DRAM WR Cycle Time**
° Every DRAM access begins at:
  - The assertion of the RAS_L
  - 2 ways to write: early or late v. CAS

---

**DRAM Write Timing**
° Every DRAM access begins at:
  - The assertion of the RAS_L
  - 2 ways to write: early or late v. CAS

---

**Row and column addresses share the same pins (A)**

- RAS_L goes low: Pins A are latched in as row address
- CAS_L goes low: Pins A are latched in as column address
- RAS/CAS edge-sensitive

---

**Control Signals (RAS_L, CAS_L, WE_L, OE_L) are all active low**

- WE_L is asserted (Low), OE_L is disasserted (High)
  - D serves as the data input pin
- WE_L is disasserted (High), OE_L is asserted (Low)
  - D is the data output pin

---

**Row and column addresses share the same pins (A)**

- RAS_L goes low: Pins A are latched in as row address
- CAS_L goes low: Pins A are latched in as column address
- RAS/CAS edge-sensitive
### Key DRAM Timing Parameters

- **$t_{RAC}$**: minimum time from RAS line falling to the valid data output.
  - Quoted as the speed of a DRAM
  - A fast 4Mb DRAM $t_{RAC}$ = 60 ns

- **$t_{RC}$**: minimum time from the start of one row access to the start of the next.
  - $t_{RC}$ = 110 ns for a 4Mb DRAM with a $t_{RAC}$ of 60 ns

- **$t_{CAC}$**: minimum time from CAS line falling to valid data output.
  - 15 ns for a 4Mb DRAM with a $t_{RAC}$ of 60 ns

### DRAM Performance

- A 60 ns ($t_{RAC}$) DRAM can
  - perform a row access only every 110 ns ($t_{RC}$)
  - perform column access ($t_{CAC}$) in 15 ns, but time between column accesses is at least 35 ns ($t_{PC}$).
    - In practice, external address delays and turning around buses make it 40 to 50 ns

- These times do not include the time to drive the addresses off the microprocessor nor the memory controller overhead.
  - Drive parallel DRAMs, external memory controller, bus to turn around, SIMM module, pins...
  - 180 ns to 250 ns latency from processor to memory is good for a “60 ns” ($t_{RAC}$) DRAM

### Something new: Structure of Tunneling Magnetic Junction

- **Tunneling Magnetic Junction RAM (TMJ-RAM)**
  - Speed of SRAM, density of DRAM, non-volatile (no refresh)
  - “Spintronics”: combination quantum spin and electronics
  - Same technology used in high-density disk-drives

### Main Memory Performance

- **Simple**:
  - CPU, Cache, Bus, Memory same width

- **Interleaved**:
  - CPU, Cache, Bus 1 word: Memory N Modules (4 Modules): example is word interleaved

- **Wide**:
  - CPU/Mux 1 word; Mux/Cache, Bus, Memory N words (Alpha: 64 bits & 256 bits)
Main Memory Performance

- DRAM (Read/Write) Cycle Time >> DRAM (Read/Write) Access Time
  - Why?

- DRAM (Read/Write) Cycle Time:
  - How frequent can you initiate an access?
  - Analogy: A little kid can only ask his father for money on Saturday

- DRAM (Read/Write) Access Time:
  - How quickly will you get what you want once you initiate an access?
  - Analogy: As soon as he asks, his father will give him the money

- DRAM Bandwidth Limitation analogy:
  - What happens if he runs out of money on Wednesday?

Increasing Bandwidth - Interleaving

Access Pattern without Interleaving:

Access Pattern with 4-way Interleaving:

Increasing Bandwidth - Interleaving

- Access Pattern without Interleaving:
  - Start Access for D1
  - Start Access for D2

- Access Pattern with 4-way Interleaving:
  - Access Bank 0
  - Access Bank 1
  - Access Bank 2
  - Access Bank 3

Main Memory Performance

- Timing model
  - 1 to send address,
  - 4 for access time, 10 cycle time, 1 to send data
  - Cache Block is 4 words
  - Simple M.P. = 4 x (1+10+1) = 48
  - Wide M.P. = 1 + 10 + 1 = 12
  - Interleaved M.P. = 1+10+1 + 3 =15

Independent Memory Banks

- How many banks?
  - number banks \geq number clocks to access word in bank
  - For sequential accesses, otherwise will return to original bank before it has next word ready
  - Prime number of banks: good for a variety of access patterns

- Increasing DRAM => fewer chips => harder to have banks
  - Growth bits/chip DRAM : 50%-60%/yr
  - Nathan Myrvold M/S: mature software growth (33%/yr for NT) - growth MB/$ of DRAM (25%-30%/yr)
Fewer DRAMs/System over Time

(from Pete MacWilliams, Intel)

<table>
<thead>
<tr>
<th>DRAM Generation</th>
<th>86</th>
<th>89</th>
<th>92</th>
<th>96</th>
<th>99</th>
<th>02</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory per</td>
<td>4 MB</td>
<td>8 MB</td>
<td>16 MB</td>
<td>64 MB</td>
<td>256 MB</td>
<td>1 Gb</td>
</tr>
</tbody>
</table>

Minimum PC Memory Size

<table>
<thead>
<tr>
<th>Memory per</th>
<th>4 MB</th>
<th>8 MB</th>
<th>16 MB</th>
<th>32 MB</th>
<th>64 MB</th>
<th>128 MB</th>
<th>256 MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>System growth</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Growth</td>
<td>@ 25%-30% / year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fast Page Mode Operation

- Regular DRAM Organization:
  - N rows x N column x M-bit
  - Read & Write M-bit at a time
  - Each M-bit access requires a RAS / CAS cycle

- Fast Page Mode DRAM
  - N x M “SRAM” to save a row

- After a row is read into the register
  - Only CAS is needed to access other M-bit blocks on that row
  - RAS_L remains asserted while CAS_L is toggled

Key DRAM Timing Parameters

- \( t_{RAC} \): minimum time from RAS line falling to the valid data output.
  - Quoted as the speed of a DRAM
  - A fast 4Mbit DRAM \( t_{RAC} = 60 \text{ ns} \)

- \( t_{RC} \): minimum time from the start of one row access to the start of the next.
  - \( t_{RC} = 110 \text{ ns} \) for a 4Mbit DRAM with a \( t_{RAC} \) of 60 ns

- \( t_{CAC} \): minimum time from CAS line falling to valid data output.
  - 15 ns for a 4Mbit DRAM with a \( t_{RAC} \) of 60 ns

- \( t_{PC} \): minimum time from the start of one column access to the start of the next.
  - 35 ns for a 4Mbit DRAM with a \( t_{RAC} \) of 60 ns

What does “Synchronous” RAM mean?

- Take basic RAMs (SRAM and DRAM) and add clock:
  - Gives SSRAM or SDRAM (Synchronous SRAM/DRAM)
  - Addresses and Control set up ahead of time, clock edges activate

- More complicated, on-chip controller
  - Operations synchronized to clock
    - So, give row address one cycle
    - Column address some number of cycles later (say 2)
    - Data comes out later (say 2 cycles later)
  - Burst modes
    - Typical might be 1, 2, 4, 8, or 256 length burst
    - Thus, only give RAS and CAS once for all of these accesses
    - Multi-bank operation (on-chip interleaving)
      - Lets you overlap startup latency (5 cycles above) of two banks

- Careful of timing specs!
  - 10ns SDRAM may still require 50ns to get first data!
  - 50ns DRAM means first data out in 50ns
Example: SDRAM timing for Lab6

- Micron 128M-bit dram (using 2Megx16bitx4bank ver)
  - Row (12 bits), bank (2 bits), column (9 bits)

DRAMs over Time

<table>
<thead>
<tr>
<th>DRAM Generation</th>
<th>1st Gen. Sample</th>
<th>‘84</th>
<th>‘87</th>
<th>‘90</th>
<th>‘93</th>
<th>‘96</th>
<th>‘99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Size</td>
<td>1 Mb</td>
<td>4 Mb</td>
<td>16 Mb</td>
<td>64 Mb</td>
<td>256 Mb</td>
<td>1 Gb</td>
<td></td>
</tr>
<tr>
<td>Die Size (mm²)</td>
<td>55</td>
<td>85</td>
<td>130</td>
<td>200</td>
<td>300</td>
<td>450</td>
<td></td>
</tr>
<tr>
<td>Memory Area (mm²)</td>
<td>30</td>
<td>47</td>
<td>72</td>
<td>110</td>
<td>165</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>Memory Cell Area (µm²)</td>
<td>28.84</td>
<td>11.1</td>
<td>4.26</td>
<td>1.64</td>
<td>0.61</td>
<td>0.23</td>
<td></td>
</tr>
</tbody>
</table>

(from Kazuhiro Sakashita, Mitsubishi)

DRAM History

- DRAMs: capacity +60%/yr, cost –30%/yr
  - 2.5X cells/area, 1.5X die size in -3 years
- ‘97 DRAM fab line costs $1B to $2B
  - DRAM only: density, leakage v. speed
- Rely on increasing no. of computers & memory per computer (60% market)
  - SIMM or DIMM is replaceable unit
  => computers use any generation DRAM
- Commodity, second source industry
  => high volume, low profit, conservative
  - Little organization innovation in 20 years
  page mode, EDO, Synch DRAM
- Order of importance: 1) Cost/bit 1a) Capacity
  - RAMBUS: 10X BW, +30% cost => little impact

DRAM Design Goals

- Reduce cell size 2.5, increase die size 1.5
- Sell 10% of a single DRAM generation
  - 6.25 billion DRAMs sold in 1996
- 3 phases: engineering samples, first customer ship(FCS), mass production
  - Fastest to FCS, mass production wins share
- Die size, testing time, yield => profit
  - Yield >> 60%
  (redundant rows/columns to repair flaws)
Today’s Situation: DRAM

- Commodity, second source industry
  - high volume, low profit, conservative
    - Little organization innovation (vs. processors)
      in 20 years: page mode, EDO, Synch DRAM

- DRAM industry at a crossroads:
  - Fewer DRAMs per computer over time
    - Growth bits/chip DRAM: 50%-60%/yr
    - Nathan Myrvold M/S: mature software growth
      (33%/yr for NT) - growth MB/$ of DRAM (25%-30%/yr)
  - Starting to question buying larger DRAMs?

Today’s Situation: DRAM

- Intel: 30%/year since 1987; 1/3 income profit

Today’s Situation: DRAM

- Summary:
  - Two Different Types of Locality:
    - Temporal Locality (Locality in Time): If an item is referenced, it will tend
      to be referenced again soon.
    - Spatial Locality (Locality in Space): If an item is referenced, items
      whose addresses are close by tend to be referenced soon.
  - SRAM is fast but expensive and not very dense:
    - 6-Transistor cell (no static current) or 4-Transistor cell (static current)
    - Does not need to be refreshed
    - Good choice for providing the user FAST access time.
    - Typically used for CACHE
  - DRAM is slow but cheap and dense:
    - 1-Transistor cell (+ trench capacitor)
    - Must be refreshed
    - Good choice for presenting the user with a BIG memory system
    - Both asynchronous and synchronous versions
    - Limited signal requires “sense-amplifiers” to recover