CS 451 / 551 / ECE 541

ADVANCED
COMPUTER ARCHITECTURE

SESSION no. 24

University of Idaho

CISC vs RISC architectures

- RISC became dominant
- CISC
  - Compilers didn't use advanced instructions (used ~ 10%)
  - Hardware implementation was complex
- RISC
  - Eliminate complex instructions
  - Make the 10% really efficient
    - Pipelining (Instr.)
    - Multiple execution units

University of Idaho

Complex instruction

```
for (i=0; i<1000; i++)
    x[i] = y[i]   // copy y to x
MOV (R1)++, (R2)++
BEQ ...
```

COMPILERS VS HAND CODING

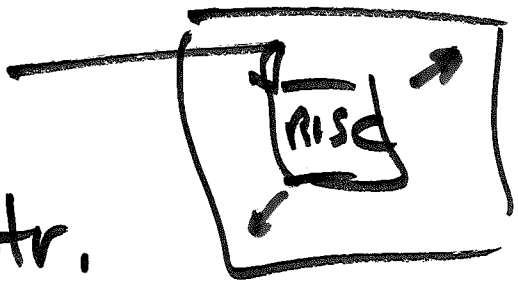
. PROGRAMMER PRODUCTIVITY

~~HIS~~. HARDWARE (PROCESS ~~ABET~~ STATE OF THE ART

MICROCODE

COMPILER → ASSEMBLY → MICROCODE
C CISC RISC

MOV (R1)++, (R2)++



CISC - 10 + clocks / instr.
RISC - 1 - 1/10 clock / instr

TODAY NOT 1.00 µm TRANSISTORS
BUT 10 nm TRANSISTORS

COMPILERS ARE SMARTER
COMPLEX INSTRUCTIONS CREEPING BACK IN.

University of Idaho

VECTOR PROCESSORS & GPU'S (GPGPU's)

- MARKET DEMANDS FOR PERFORMANCE ARE HIGH
  - MOORE'S LAW FREQUENCY SCALING STALLED.
- NEW SOFTWARE TOOLS - MAKE IT EASIER
  - OLD: GNU - C, C++, LIBS, ...
  - NEW: CLANG: C, C++, C#, OBJEC. C.,

UNIFY

- VECTOR LIBRARIES,
- GRAPHICS LIBS
- THREAD LIBS

EASIER FOR PROGRAMMERS

CACHE COHERENCE

- CONSISTENT VALUES IN SEPARATE CACHES

BUS SNOOPING

- ALL CPU'S WATCH MEMORY BUS
  - IF SOMEONE ELSE WRITES TO A CACHE BLOCK, THAT THEY ALSO HAVE A COPY OF, "INVALIDATE" CPU'S COPY,
  - NEXT ACCESS - GET FRESH COPY
- PRO - FAST
- CON - NOT VERY EXTENSIBLE 4, 8, ..

DIRECTORY

- PRO - EXTENSIBLE
- CON - SLOWER — 1,000

Signal Integrity for Digital Systems

Design the system to enforce the digital abstraction

- Logical '0' is 0
- Logical '1' is 1
- Nothing in between

Topics

1. Signal standards
2. Interconnect
3. Power integrity & power distribution
4. General recommendations

Three views of signals

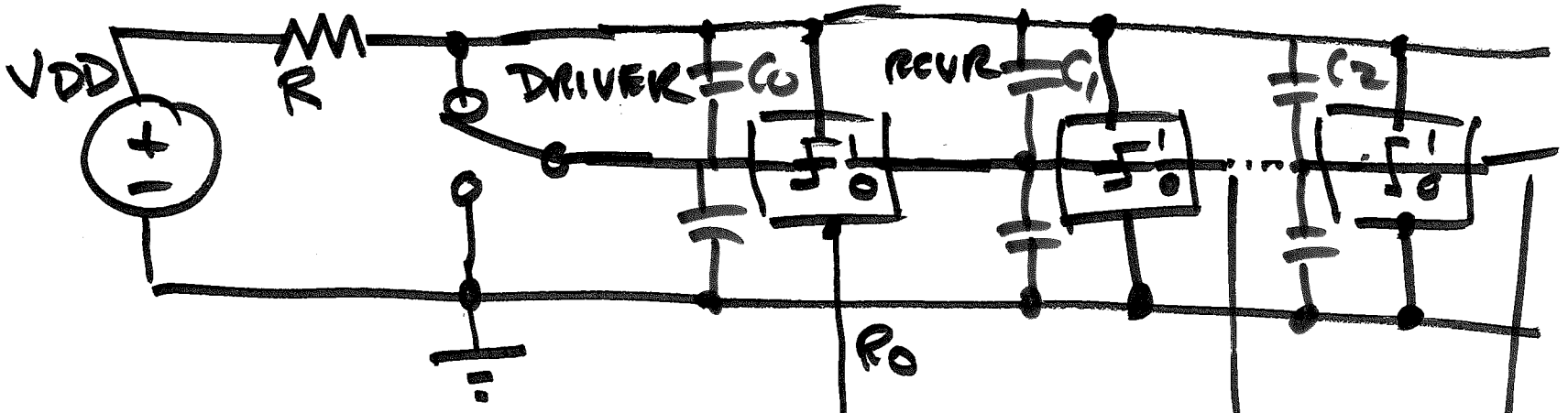
1. Ideal: Logic design "textbook" behavior

2. Lumped parameter behavior:
   - Slower rise times, Tr > 1/10 pulse transit time
   - Lump inductances & capacitances
   - Use circuit models

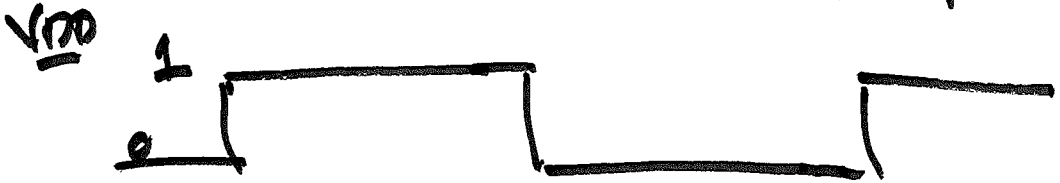
3. Wave behavior:
   - Faster rise times, Tr < 1/10 pulse transit time
   - Signal paths act like waveguides
   - Reflections become an issue

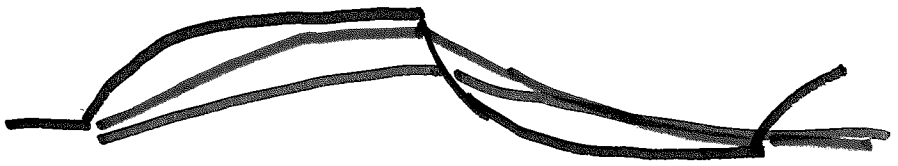
University of Idaho

WARNING: ELECTRONIC CIRCUIT CONTENT!



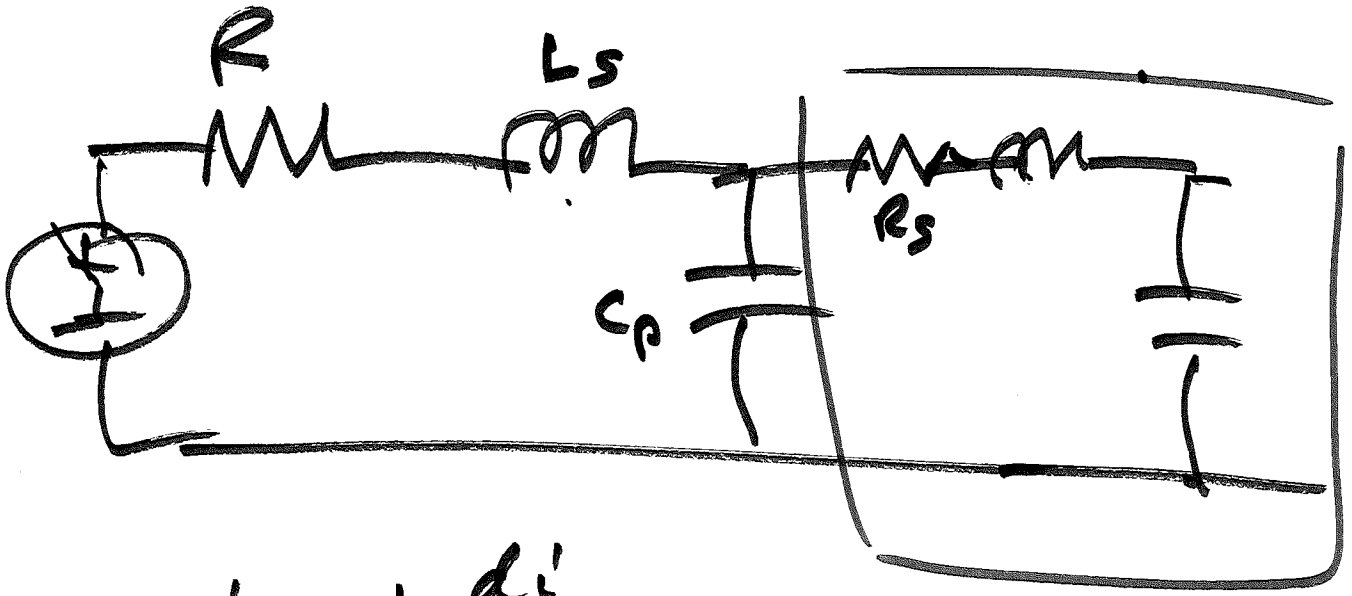
$C = C_0 || C_1 || \ldots || C_n$





$$V_R(t) = V_{DD}\left(1 - e^{-t/RC}\right)$$

(GROUND) BOUNCE + POWER SUPPLY DROOP



$$V_L = L \frac{di}{dt}$$

Assume: $\frac{di}{dt} \cong 1 \frac{mA}{ns}$

$$V_L = L \times \frac{10^{-6}}{10^{-9}} = = 1000 \times V_L$$

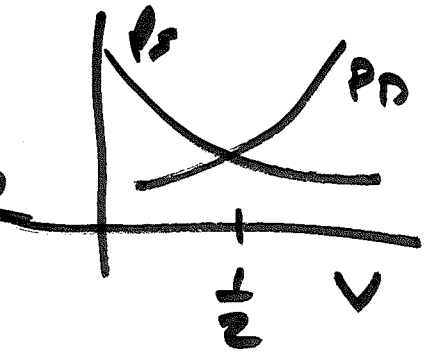
TRENDS:

VOLTAGES: 5V → 3.3 → 2.5 → 1.8 - 1.1 - 0.5

SINGLE ENDED SIGNALLING

V_{DD} : 0.5V => $P_S = P_D$

$P_D \propto C_L f V_{DD}^2$

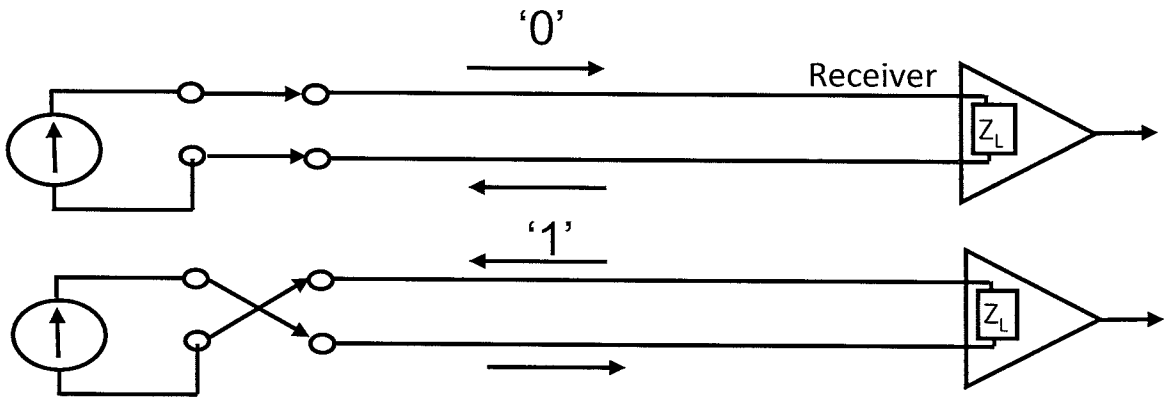


- REDUCED MARGIN FOR ERROR
- NOISE -

LOW - VOLTAGE DIFFERENTIAL SIGNALLING $\approx \frac{1}{2} V$

- SWITCH CURRENT DIRECTION, NOT VOLTAGE
- POWER INDEPENDENT OF SWITCHING
- COMMON MODE NOISE REJECTION

Differential Signaling



Action:

•Detection based on input current direction (positive or negative)

Routing:

- Each signal has its own forward and return lines.
- Can tailor with desired transmission line characteristics, field cancellation
- Minimize frequency effects and signal distortion

Advantage: Excellent common-mode noise rejection, short rise time

Disadvantage: PCB routing more complicated

Example

Low Voltage Differential Signaling (LVDS)

TRENDS IN INTERCONNECT

- LOW VOLTAGE DIFFERENTIAL
- SERIAL BUSSES

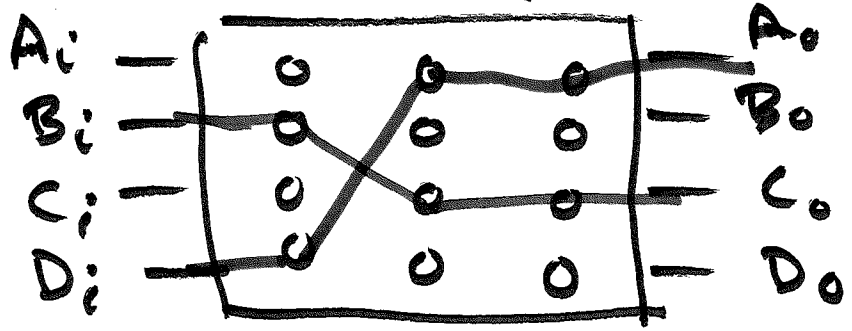
PCI - PERIPHERAL COMPONENT INTERRONNECT

PCI-X "PCI EXPRESS"

SERIAL

~~INTERCONNECT~~ (SWITCHING) INTERCONNECT NETWORKS

CROSSBAR



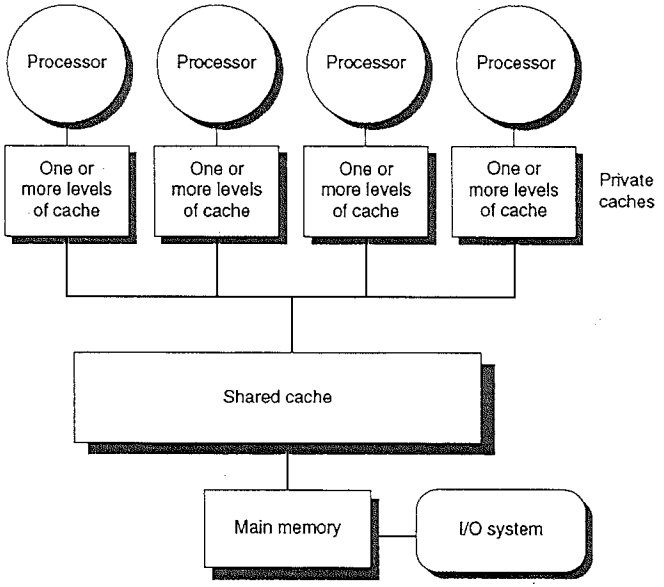
CH5 - Multiprocessors & Thread-Level Parallelism

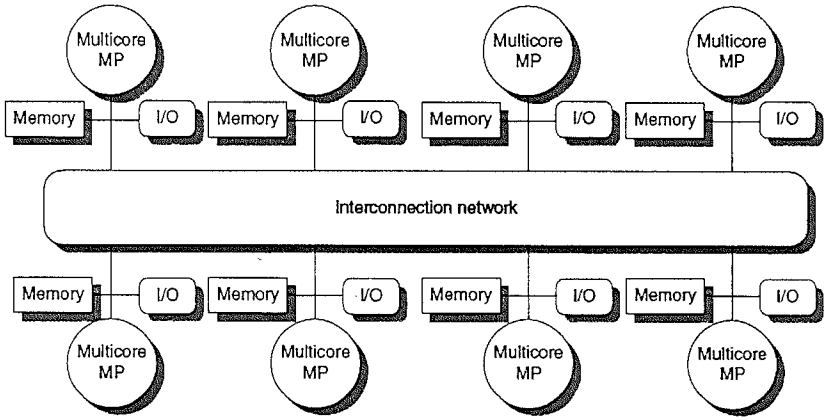
Introduction

- Thread-Level parallelism
  - Have multiple program counters
  - Uses MIMD model
  - Targeted for tightly-coupled shared-memory multiprocessors
- For *n* processors, need *n* threads
- Amount of computation assigned to each thread = grain size
  - Threads can be used for data-level parallelism, but the overheads may outweigh the benefit

Types

- Symmetric multiprocessors (SMP)
  - Small number of cores
  - Share single memory with uniform memory latency
- Distributed shared memory (DSM)
  - Memory distributed among processors
  - Non-uniform memory access/ latency (NUMA)
  - Processors connected via direct (switched) and non-direct (multi-hop) interconnection networks





Cache Coherence

- Coherence
  - All reads by any processor must return the most recently written value
  - Writes to the same location by any two processors are seen in the same order by all processors
- Consistency
  - When a written value will be returned by a read
  - If a processor writes location A followed by location B, any processor that sees the new value of B must also see the new value of A

Cache Coherence

- Processors may see different values through their caches:

Time	Event	Cache contents for processor A	Cache contents for processor B	Memory contents for location X
0				1
1	Processor A reads X	1		1
2	Processor B reads X	1	1	1
3	Processor A stores 0 into X	0	1	0