Lecture 1 (20.02 Thu.)

Computer Architecture

The transformation hierarchy

Levels of transformation

Design goals

Google tensor processing unit (TPU)

Tesla self-driving computer

Redundant cores for better safety

Energy efficiency and performance

Intel Optane persistent memory (3D-XPoint)

Cerebra's wafer scale engine

Graphics processing unit (GPU)

UPMEM processing-in-DRAM

Processing-in-memory

AI/ML chips

Reliability and security

RowHammer

DRAM row

Meltdown and Spectre

Genome sequencing

Nanopore sequencing

Computing paradigms

Accelerators (algorithm-hardware codesign)

Memory and storage systems

Bahnhof Stadelhofen

Santiago Calatrava

Architecture

Tradeoffs

Evaluation criteria

Principled design

Design constrains

Lecture 2A (21.02 Fri.)

Principled design

Machine Learning

Combinational logic

Sequential logic

Tradeoffs

Lecture 2B (21.02 Fri.)

Transformation hierarchy

Hamming distance

Error correcting codes

Levels of transformation

Abstraction levels

Accelerator

Meltdown & Spectre

Security attacks

Hardware security vulnerabilities

Speculative execution

Vonn Neuman

Microarchitecture

Instruction Set Architecture (ISA)

Cache

Timing side channel

Rowhammer

Disturbance errors

DRAM module

DRAM cell

Crosstalk effect (cell-to-cell interference)

Bit flips

Page table entry (PTE)

PARA: Probabilistic Adjacent Row Activation

Byzantine failures

Maslow’s Hierarchy

Reliability

DRAM refresh

DRAM cell

Access transistor

Bitline

Wordline

Lecture 3A (27.02 Thu.)

Mysteries in Computer Architecture

DRAM Refresh

Retention Time Profile of DRAM

Manufacturing Process Variation

RAIDR: Eliminating Unnecessary DRAM Refreshes

Bloom Filters

VRT: Variable Retention Time

Memory Performance Attacks

Many Cores on Chip

Unexpected Slowdowns in Multi-Core

Disparity in Slowdowns

Memory Controller

DRAM Bank Operation

DRAM Row / Column

Row Buffer

FR-FCFS: First Ready First Come First Served

Row-Hit-first

Oldest-first

Denial-of-Service Attacks

Row Buffer Locality

Memory-Intensive Applications

Lecture 3B (27.02 Thu.)

Lab Sessions

Grading Policy

Deadlines for Lab Exercises and Lab Reports

The Transformation Hierarchy

Hardware Prototyping

Debugging a Hardware Implementation

Hardware Description Languages (HDL)

Hardware Design Flow

Computer-Aided Design (CAD)

Project Brainwave

Amazon EC2 F1

FPGA-based DNA Sequencing

FPGA-based DRAM Characterization

SoftMC

FPGA-based Flash Memory Characterization

Basys 3 FPGA Board

High Level Summary of Labs

Seven Segment Display

Finite State Machines

ALU: Arithmetic and Logic Unit

Testing and Simulation

Assembly Language

FPGA: Field Programmable Gate Array

FPGA Building Blocks

Look-Up Tables (LUT)

Switches

Multiplexers

Xilinx Zynq Ultrascale+

FPGA Design Flow

Xilinx Vivado

Verilog code

Logic Synthesis

Placement and Routing

Lecture 4 (28.02 Fri.)

Combinational logic circuits

Transistor

Moore’s Law

HW/SW interface

Boolean algebra

Boolean equations

Logic gates

Microprocessors

FPGA

ASIC

MOS transistor

Transistor gate/source/drain

Power supply, ground

n-type/p-type MOS transistor

nMOS

pMOS

Complementary MOS (CMOS)

Boolean inverter

CMOS NOT gate

Pull-up, pull-down

CMOS NAND gate

Truth table

CMOS AND gate

Functional Specification

Boolean Algebra Axioms

Lecture 5 (05.03 Thu.)

Combinational logic circuits

Big Data

Machine learning

Genome analysis

Transistor

Moore’s Law, Dennard Scaling

Data movement bottleneck

Main memory

HW/SW interface

Boolean algebra, Boolean Equations

Logic gates

FPGA

MOS transistor

Transistor gate/source/drain

Power supply, ground

n-type/p-type MOS transistor

Complementary MOS (CMOS)

Boolean inverter

CMOS NOT gate

Pull-up, pull-down

CMOS NAND gate

Truth table

CMOS AND gate

Logical completeness

Floating value (Z)

dynamic/static power consumption

Leakage current

Lecture 6 (06.03 Fri.)

Logic minimization / simplification

Karnaugh Map (K-Map)

Bit value X: Don't care

BCD: Binary coded decimal

Sequential circuit

Circuits that can store information

Capturing data

Cross-coupled inverter

Metastable states

Storage element

Memory

SRAM (static random access memory)

DRAM (dynamic random access memory)

Flash memory, hard disk, tape, and non-volatility

Latches and flip flops

R-S (Reset-Set) Latch

Forbidden state

Gated D Latch

Register

Address

Reading from memory

Writing to memory

Wordline

Address Decoder

Multiplexer

Addressability

State

State machine diagram

Clock

Finite State Machine (FSM)

Next state logic

State register

Output logic

D Flip Flop

Master latch

Slave latch

Edge-triggered device

Rising edge

Falling edge

4-bit register

Moore machine

Mealy machine

Transition diagram

State encoding

Lecture 7A (12.03 Thu.)

Sequential circuit

Finite State Machine

Flip flop

State transition table

FSM state encoding

Binary encoding

One-hot encoding

Output encoding

Moore and Mealy FSMs

State transition diagram

LC-3 processor

Lecture 7B (12.03 Thu.)

Hardware Description Language (HDL)

Verilog

VHDL

Hierarchical design

Modules

Top-down / Bottom-up design methodologies

Top-level module, sub-module, leaf cell

Bus

Manipulating Bits

Bit slicing

Concatenation

Duplication

Behavioral HDL

Structural (gate-level) description

Behavioral / functional description

Bitwise operators

Reduction operators

Conditional assignments

Precedence of operators

Tri-state buffer

Synthesis

Simulation

Gate-level implementation

Parametrized modules

Sequential Logic in Verilog

Always block

Sensitivity list

Posedge

D Flip-Flop

Blocking assignment

Non-blocking assignment

Asynchronous/Synchronous reset

Blocking/Non-blocking assignment

Glitches

Case statement

Implementing FMS

Lecture 8 (13.03 Fri.)

Area

Speed / Throughput

Power / Energy

Design time

Circuit timing

Combinational circuit timing

Combinational circuit delay

Contamination delay

Propagation delay

Longest / Shortest path

Critical path

Glitch

Fixing glitches with K-map

Sequential circuit timing

D flip-flop

Setup / Hold / Aperture time

Metastability

Non-deterministic convergence

Contamination delay clock-to-q

Propagation delay clock-to-q

Correct sequential operation

Hold time constraint

Timing analysis

Clock skew

Safe timing

Circuit verification

High level design

Circuit level

Functional equivalence

Functional tests

Timing constraints

Functional verification

Testbench

Device under test (DUT)

Simple / Self-checking / Automatic testbench

Wavefront diagrams

Clock generation

Golden model

Timing verification

Timing report / summary

Lecture 9 (19.03 Thu.)

Basic elements of a computer

The von Neumann Model

Addressability

Address Space

Word-Addressable Memory

MIPS

LC-3

MIPS memory

Unique address

Byte-addressable

Big endian vs Little endian

MAR and MDR

Load and store instructions

Processing unit

Arithmetic and Logic Unit (ALU)

Registers

Input and Output (IO)

Control Unit

Programmer Visible State

Instructions

Program Counter

Sequential Execution

Memory

Instruction set Architecture (ISA)

Instruction Format

R-type

Operate instructions

Load/Store Word

I-Type instruction

The Instruction Cycle

Fetch phase

Instruction register

Decode phase

Fetch operands

Jump

Unconditional branch or jump

Base register

The Instruction Set

Operand

Lecture 10a (20.03 Fri.)

Von-Neumann Model

Instruction Set Architecture (ISA)

MIPS

LC-3

LC-3b

Assembly Language

Single-Cycle Microarchitecture

Addressing Mode

Instruction

Operate instruction

Movement instruction

Control Flow instruction

Addressability

Word-addressable

Byte-addressable

Address Space

Memory Address Register

Memory Data Register

Arithmetic and Logical Unit (ALU)

General Purpose Register

Register File

Function Return Value

Function Argument

Stack Pointer

Frame Pointer

Function Return Address

Instruction Register

Instruction Pointer / Program Counter

Memory Load

Memory Store

Instruction Cycle

Opcode

Operand

Semantic Gap

Immediate Operand

Register Operand

Memory Addressing Mode

Lecture 10b (20.03 Fri.)

Instruction Set Architecture (ISA)

LC-3

MIPS

Assembly

Von Neumann model

Instruction cycle

Instruction

Operate instruction

Data movement instruction

Control flow instruction

Unary/binary operation

Literal or immediate

Addressing mode

PC-relative addressing mode

Indirect addressing mode

Base+offset addressing mode

Immediate addressing mode

Source/destination register

Machine code

Conditional branch

Jump

Condition codes

Loop

LC-3 data path

Assembly programming

Programming constructs

Sequential construct

Conditional construct

Iterative construct

OS service call

End Of Text (EOT)

Sentinel

Debugging

Interactive debugging

Breakpoint

If-else statement

While loop

For loop

Arrays in MIPS

Function call

Caller/callee

Arguments/return value

Stack

Preserved/nonpreserved registers

Lecture 11 (26.03 Thu.)

Microarchitecture

Von Neumann Machine

ISA

Stored program computer

Sequential instruction processing

Unified memory

Instruction pointer

Data flow model

Data flow dependence

Instruction pointer

Data flow node

Control-flow execution order

Data-flow execution order

Pipeline

Instruction and data caches

General purpose registers

Virtual ISA

Single-cycle microarchitecture

Multi-cycle microarchitecture

Critical path

Control unit

Instruction Fetch

Instruction Decode

Functional units

Datapath

Control logic

CPI

Register file

ALU (Arithmetic Logic Unit)

Store writeback

Arithmetic and Logical instructions

Instruction types (R-type, I-type, J-type)

MUX (Multiplexer)

Source/destination register

Immediate value

Jump instruction

Conditional Branch

Lecture 12 (27.03 Fri.)

ALU: Arithmetic-Logic Unit

Single-cycle MIPS Datapath

Control signals

Datapath configuration

R-type, I-type, LW, SW, Branch, and Jump datapath configurations

Control logic

Hardwired control (combinational)

Sequential/Microprogrammed control

Performance analysis

CPI: Cycles per Instruction

Critical path

Slowest instruction

Execution time of an instruction / of a program

Single cycle microarchitecture complexity

Fetch, decode, evaluate address, fetch operands, execute, store result

Magic memory

Instruction memory and data memory

REP MOVS and INDEX instructions

Microarchitecture design principles

Bread and butter (common case) design and Amdahl's law

Balanced Design

Key system design principles: keep it simple, keep it low cost

Multi-cycle critical path

Multi-cycle microarchitecture

Multi-cycle performance

Overhead of register setup/hold times

Main controller FSM

Lecture 13 (02.04 Thu.)

Pipelining

Control & data dependence handling

State maintenance and recovery

Multi-cycle design

Concurrency

Instruction throughput

Assembly line processing

Pipeline stages

Identical operations

Independent operations

Uniformly partitionable suboperations

The Instruction processing cycle

Instruction fetch (IF)

Instruction decode and Register operand fetch (ID/RF)

Execute/Evaluate memory address (EX/AG)

Memory operand fetch (MEM)

Store/writeback result (WB)

Pipeline registers

Control points

Control signals

Pipeline stalls

Resource contention

Dependences (data/control)

Long-latency (multi-cycle) operations

Data dependences

Flow dependence

Output dependence

Anti dependence

Data dependence handling

Interlocking

Scoreboarding

Combinational dependence check logic

Data forwarding/bypassing

RAW dependence handling

Stalling hardware

Lecture 14 (03.04 Fri.)

Data dependences

Stalling

Stalling hardware

Hazard unit

Control dependences

Branch misprediction penalty

Instructions flushing

Early branch resolution

Data forwarding

Branch prediction

Pipelined performance

SPECINT2006 benchmark

Average CPI

Software-based interlocking

Hardware-based interlocking

Pipeline bubbles

Software-based instruction scheduling

Hardware-based instruction scheduling

Static / dynamic scheduling

Variable-length operation latency

Profiling

Multi-cycle execution

Exceptions

Interrupts

Precise exceptions / interrupts

Instruction retiring

Exception handling

Precise exceptions in pipelining

Reorder buffer (ROB)

ROB entry

Content Addressable Memory (CAM)

Register renaming

Architectural register ID

Physical register ID

Output dependences

Anti dependences

In-order pipeline with ROB

Lecture 15a (09.04 Thu.)

In-order pipeline

Stalling

Latency

Dispatch Stalls

In-order dispatch

Out-of-order dispatch

Reservation stations

Out-of-order (OoO)

Out-of-order execution

Functional unit (FU)

Tomasulo's Algorithm

Lecture 15b (09.04 Thu.)

OoO: Out of Order Execution

Tomasulo's algorithm

Register alias table (RAT)

Physical register file (PRF)

Tag/value broadcast

Reservation station

Instruction scheduling/dispatching

Instruction window

Dataflow graph

Precise exceptions

Frontend register file

Architecture register file

Reorder buffer (ROB)

Register renaming

Latency tolerance

Instruction window size

Memory disambiguation / unknown address problem

Store - Load dependency

LQ/SQ: Load Queue / Store Queue

Data forwarding between stores and loads

Lecture 16a (23.04 Thu.)

Dataflow graph

Precise exceptions

Frontend register file

Architecture register file

Reorder buffer (ROB)

Register renaming

Latency tolerance

Instruction window size

Memory disambiguation / unknown address problem

Store - Load dependency

LQ/SQ: Load Queue / Store Queue

Data forwarding between stores and loads

Lecture 16b (23.04 Thu.)

Superscalar execution

Dataflow

Out-of-order

Irregular parallelism

Von Neumann model

Precise state

Parallelism control

Bookkeeping overhead

Multiple instructions per cycle

N-wide superscalar

In-order superscalar

Dependency checking

Lecture 17 (24.04 Fri.)

Direction predictor

Branch target buffer

Always taken / not taken

Backward taken, forward not taken

Profile based

Program analysis based

Last time prediction

Two bit counter based prediction

Two level prediction

Hybrid branch prediction

Perceptron based branch prediction

Tag and BTB index

Branch history table

Hysteresis

Bimodal prediction

Two-level adaptive training branch prediction

Global branch correlation

Two-level global branch prediction

Pattern history table

Global history register

Global predictor accuracy

Alpha 21264 Tournament Predictor

Local and global prediction

Loop branch detector and predictor

Perceptron based branch predictor

Hybrid history length based predictor

Prediction function

Training function

Branch confidence estimation

Handling control dependencies

Delayed branching

Fancy delayed branching

Lecture 18a (30.04 Thu.)

Very Long Instruction Word (VLIW)

Superscalar

Lock-Step Execution

RISC

Commercial VLIW Machines

Multiflow TRACE

Intel IA-64

Cydrome Cydra 5

Transmeta Crusoe

TI C6000, Trimedia, STMicro

VLIW Tradeoffs

Superblock

IMPACT

Lecture 18b (30.04 Thu.)

Systolic Arrays

Processing Element (PE)

Regular array of PEs

Convolution

LeNet-5

AlexNet

GoogLeNet

ResNet

Two-Dimensional Systolic Arrays

Combinations

Programmability in Systolic Arrays

Staged execution

Pipeline-Parallel (Pipelined) Programs

Stages of Pipelined Programs

The WARP Computer

TPU

Decoupled Access/Execute (DAE)

Decoupled Access/Execute

Astronautics ZS-1

Loop Unrolling

Pentium 4

Lecture 18c (30.04 Thu.)

Fine-Grained Multithreading

CDC 6600

Heterogeneous Element Processor (HEP)

Denelcor HEP

Fine-Grained Multithreading in HEP

Multithreaded Pipeline

Sun Niagara Multithreaded Pipeline

Modern GPUs are FGMT Machines

NVIDIA GeForce GTX 285

The Tera Computer System

Lecture 19 (07.05 Thu.)

SIMD processing

GPU

Regular parallelism

Single Instruction Single Data (SISD)

Single Instruction Multiple Data (SIMD)

Multiple Instruction Single Data (MISD)

Systolic array

Streaming processor

Multiple Instruction Multiple Data (MIMD)

Multiprocessor

Multithreaded processor

Data parallelism

Array processor

Vector processor

Very Long Instruction Word (VLIW)

Vector register

Vector control register

Vector length register (VLEN)

Vector stride register (VSTR)

Prefetching

Vector mask register (VMASK)

Vector functional unit

CRAY-1

Seymour Cray

Memory interleaving

Memory banking

Vector memory system

Scalar code

Vectorizable loops

Vector chaining

Multi-ported memory

Vector stripmining

Gather/Scatter operations

Masked vector instructions

Lecture 20 (8.05 Fri.)

Memory Banking

Vector instruction execution

Vector length

Vector instruction level parallelism

SIMD processing

CRAY

SIMD

Vector processing

Automatic Code Vectorization

Vectorized Code

Scalar Sequential Code

Amdahl's Law

Regular data level parallelism

Vectorizability of code

ISAs including SIMD operations

Modern ISAs

MMX operations

Packed arithmetic

Fine-grained multithreading

Multithreaded pipeline

Warps

GPUs and SIMD engines

Programming using threads

Hardware Execution Model

Exploiting parallelism

SISD

MIMD

SPMD on SIMT Machine

Sequential instruction stream

Multiple instruction streams

Scalar instructions

Fine grained multithreading of warps

Warp-Level FGMT

Warp Execution

SIMT Memory Access

Warp Instruction Level Parallelism

CPU threads and GPU kernels

GPU SIMT Code

CUDA code

Blocks to Warps

Streaming Multiprocessors (SM)

Streaming Processors (SP)

NVIDIA Fermi architecture

Warp-based vs Traditional SIMD

Dynamic Warp Formation

Two-Level Warp Scheduling

Branch divergence

Long latency operations

Sub-warps

Two-Level Round Robin

NVIDIA GeForce GTX 285

NVIDIA V100

Lecture 21a (14.05 Thu.)

Memory

Computation

Communication

Storage/Memory

Virtual memory

Physical memory

Abstraction layers

Load/store data

Flip-flops (latches)

Random Access Memory (RAM)

Static RAM (SRAM)

Dynamic RAM (DRAM)

Storage technology (flash memory, hard disk, tape)

Memory array

Decoder

Wordline

Memory bank

Sense amplifier

Charge loss

Refresh

Lecture 21b (14.05 Thu.)

DRAM vs SRAM

Memory hierarchy

Memory locality

Temporal locality

Spatial locality

Caching basics

Caching in a pipelined design

Hierarchical latency analysis

Access latency and miss penalty

Hit-rate, miss-rate

Direct-mapped cache

Set associativity

Full associativity

Eviction/replacement policy

Last recently used (LRU)

Random

Lecture 22 (15.05 Fri.)

DRAM

Memory hierarchy

Caching

Temporal locality

Spatial locality

Cache Line/Block

Cache hit/miss

Placement

Replacement

Granularity of management

Write policy

Tag Store

Data store

Average Memory Access Time (AMAT)

Direct map cache

Conflict miss

Set associativity

Full associativity

Degree of associativity

Capacity miss

Eviction/Replacement policy

LRU, MRU, Random replacement policies

Set thrashing

Write-back

Write-through

Subblocked (Sectored) Caches

Instruction cache

Data cache

Multi-level caching

Compulsory misses

Lecture 23a (22.05 Fri.)

Cache structure

Tag store

Data Store

Bookkeeping

Cache performance

Cache size

Block size

Associativity

Replacement policy

Insertion/Placement policy

Hit/Miss rate

Hit/Miss latency/cost

Data access patterns

Data layout

Column major and Row major data layouts

Tiling and blocking

Multi-core issues in caching

Shared vs private caches

Performance isolation

QoS (Quality of Service), Fairness, and Starvation

Cache fragmentation

Dynamic partitioning

Shared resource view

Cache coherence

Consistency problem

Software-level coherence: Flush-Local/Global/Cache instructions

Scratchpad memory - software managed caches

Simple coherence scheme: snooping and broadcasting

Maintaining coherence

Write propagation

Write serialization

Hardware cache coherence

Snoopy bus

Directory based

Lecture 23b (22.05 Fri.)

Virtual memory

Physical memory

Infinite capacity

Relocation

Protection and isolation

Sharing

Linear address

Real address

Page table

OS-managed lookup table

Address translation

Physical frame

Demand paging

Placing, replacement, granularity of management, write policy

Page

Page size

Page offset

Page fault

Virtual page number (VPN)

Physical page number (PPN)

Virtual address

Physical address

Lecture 24 (28.05 Thu.)

Physical page number (physical frame number)

Page replacement policy

Page dirty bit

Page table base register (PTBR)

Page fault

Multi-level (hierarchical) page table

Translation Lookaside Buffer (TLB)

Memory Management Unit (MMU)

Page Table Entry

Tag store

Page hit, page fault

OS trap handler

Direct Memory Access (DMA)

Interrupt processor

Access protection bits

Access protection exception

Privilege levels

DRAM disturbance errors

RowHammer

Lecture 25 (29.05 Fri.)

Memory Latency

Runahead Execution

Power Efficiency

Read Mapping Filtering

Thing Big

Aim High

Research is about insight, not numbers