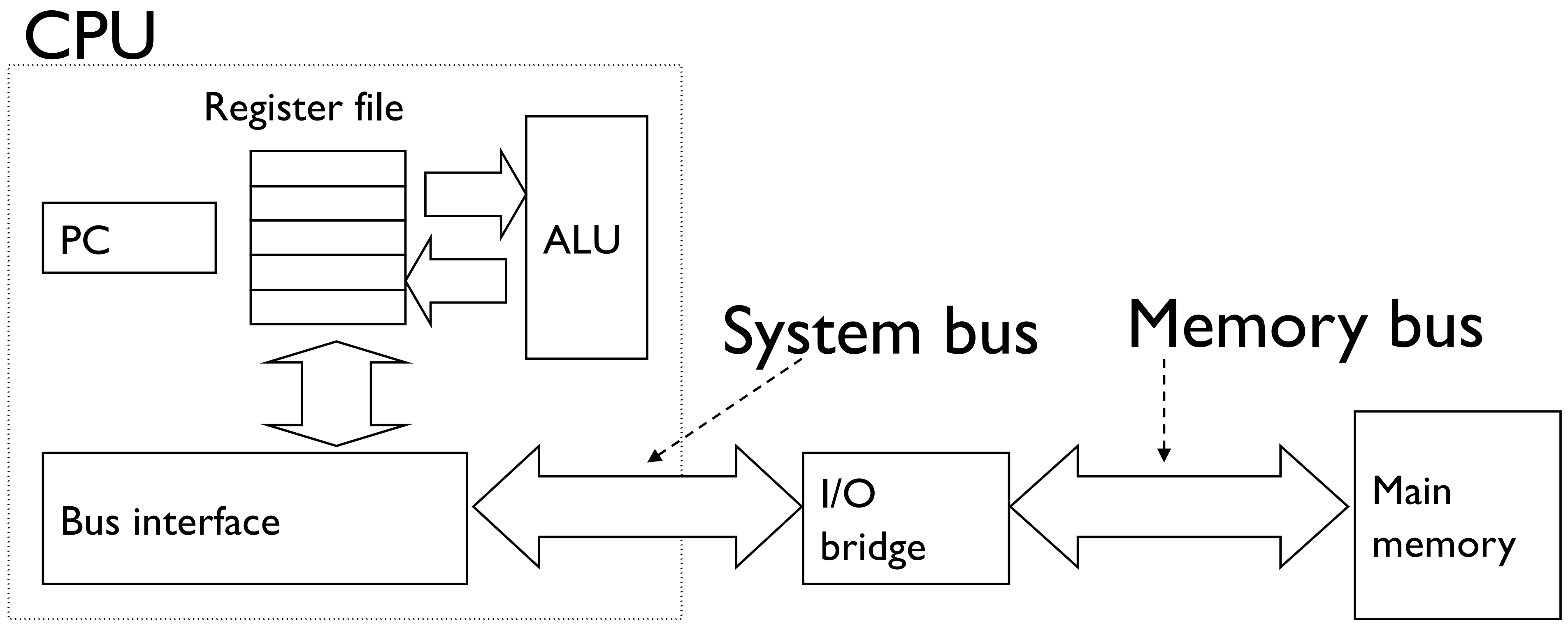
Y86 Introduction

- Topic:
  - What is the y86?
  - Why are we learning about it?
  - What is its visible state?
- Learning outcomes
  - Identify the basic components of a processor
  - Describe the y86 processor in terms of its fundamental components.
- Reading
  - Bryant and O’Halloran: Chapter 4 through section 4.1.2 & 4.1.4

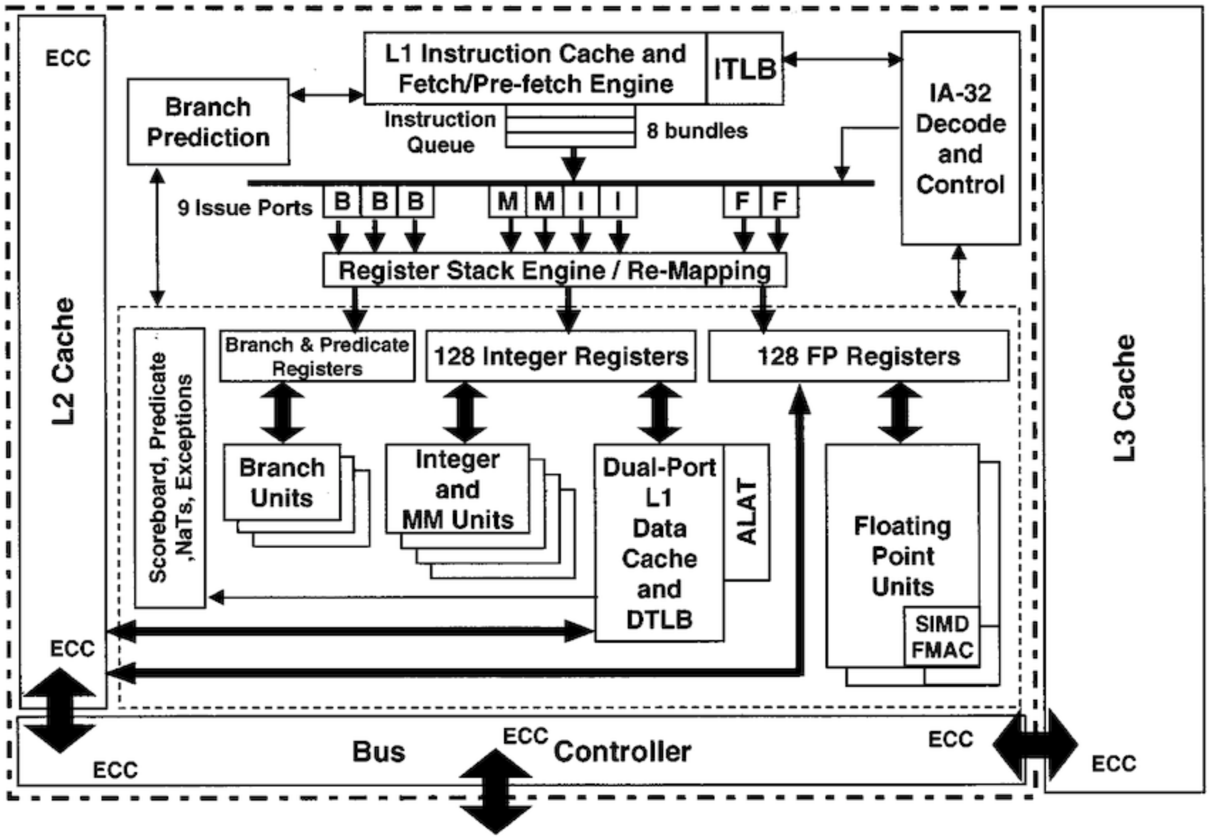
Much of this material is derived from that of Bryant and O’Halloran.

The Artist's Rendition of a CPU



From *Computer Systems: A Programmer's Perspective*

A Modern CPU


ECC
L2 Cache
Branch Prediction
L1 Instruction Cache and Fetch/Pre-fetch Engine
ITLB
Instruction Queue
8 bundles
IA-32 Decode and Control
9 Issue Ports
B
B
B
M
M
I
I
F
F
Register Stack Engine / Re-Mapping
Scoreboard, Predicate ,NaTs, Exceptions
Branch & Predicate Registers
128 Integer Registers
128 FP Registers
Branch Units
Integer and MM Units
Dual-Port L1 Data Cache and DTLB
ALAT
Floating Point Units
SIMD FMAC
ECC
ECC
Bus
ECC
Controller
ECC
ECC
L3 Cache

The y86 Processor

Registers (RF: Register File)

`%rax`	`%rsp`	`%r8`	`%r12`
`%rcx`	`%rbp`	`%r9`	`%r13`
`%rdx`	`%rsi`	`%r10`	`%r14`
`%rbx`	`%rdi`	`%r11`	

Condition Codes (CC)

ZF	SF	OF

Stat: Status Register

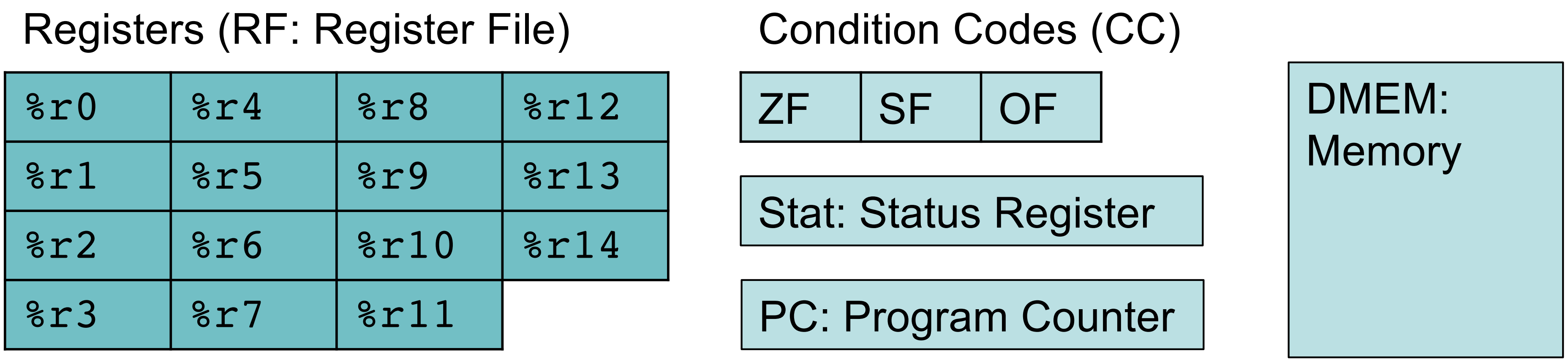
PC: Program Counter

DMEM: Memory

Program visible state

- Referenced by the ISA
- Used in assembly programs
- Actual implementation can have lots of other things

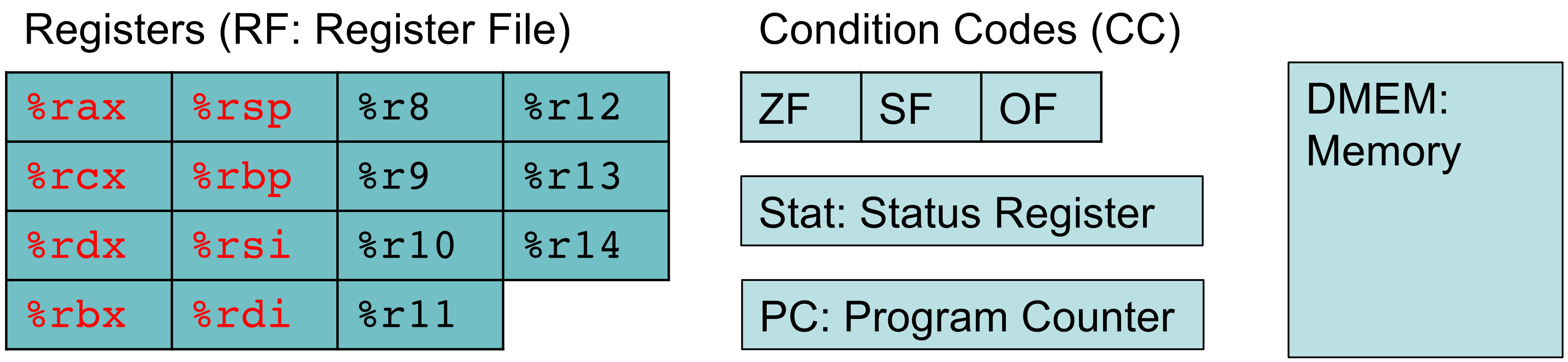
The y86 Processor – Registers



Registers:

- **High performance storage used to manipulate data**
- **Processors often execute instructions in a single cycle; accessing memory can take 10's or 100's of cycles.**
- **Most processors have a few tens of registers.**
- **The y86 has 15 64-bit registers.**

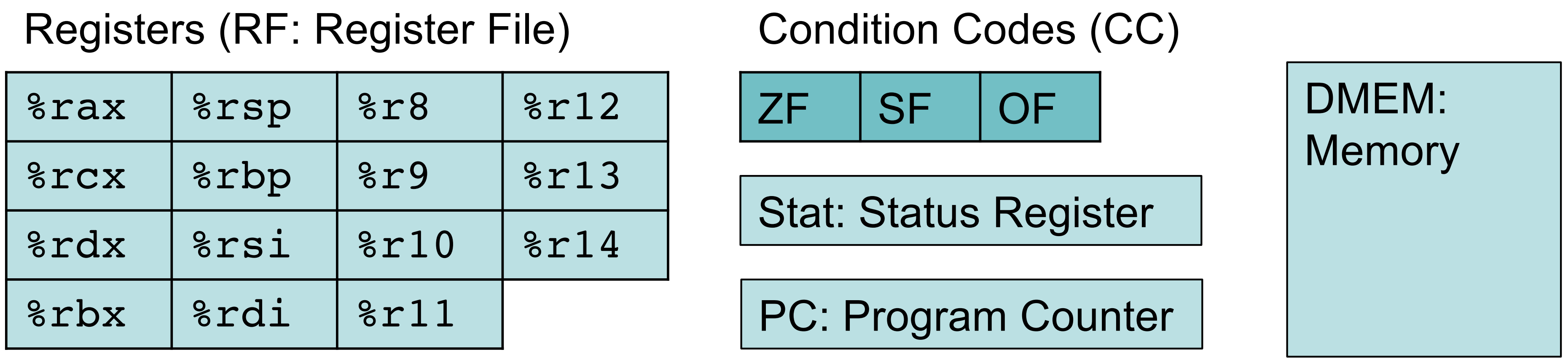
The y86 Processor – Registers



Registers:

- **Some registers have designated purposes, e.g., %r4 is als called %rsp which is used as the stack pointer.**
- **Register %r15 is invalid**

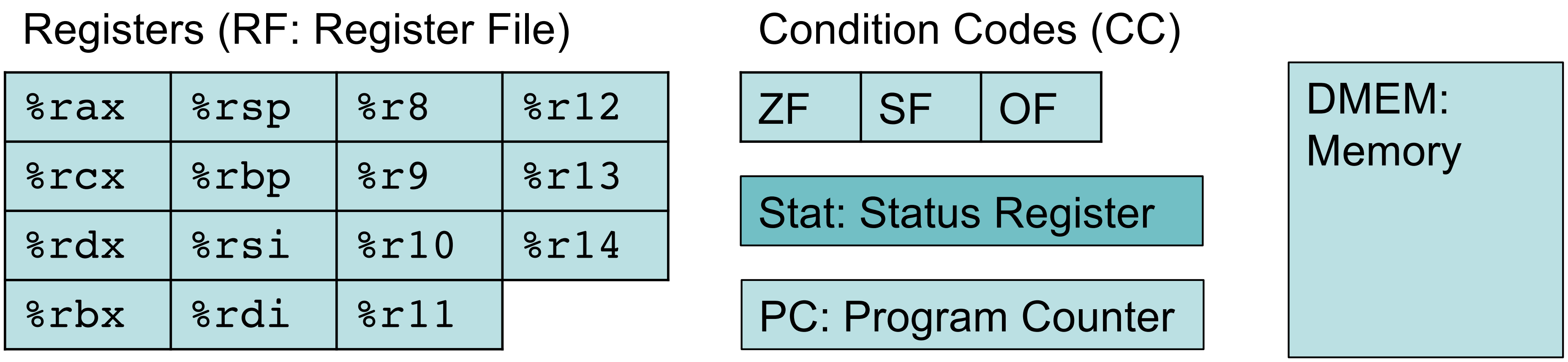
The y86 Processor – Condition Codes



Condition Codes:

- A collection of bits set by arithmetic and logical instructions (ALU operations)
- ZF: Zero-flag: The last ALU operation produced a 0
- SF: Sign-flag: The last ALU operation produced a negative number
- OF: Overflow-flag: The last ALU operation produced an overflow

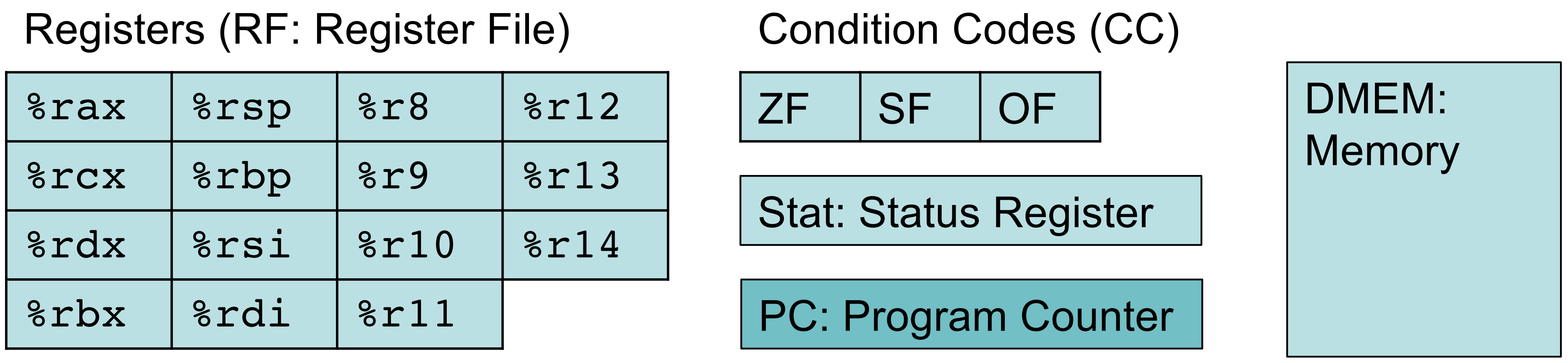
The y86 Processor – Status



Program Status Register

- Indicates normal operation or an error condition
- 1: AOK Normal Operation
- 2: HLT Halt Instruction encountered
- 3: ADR Bad address encountered
- 4: INS Invalid instruction encountered

The y86 Processor – PC



Program Counter (PC)

- Indicates the address of the **next** instruction to execute

The y86 Processor – Memory

Registers (RF: Register File)

`%rax`	`%rsp`	`%r8`	`%r12`
`%rcx`	`%rbp`	`%r9`	`%r13`
`%rdx`	`%rsi`	`%r10`	`%r14`
`%rbx`	`%rdi`	`%r11`	

Condition Codes (CC)

ZF	SF	OF

Stat: Status Register

PC: Program Counter

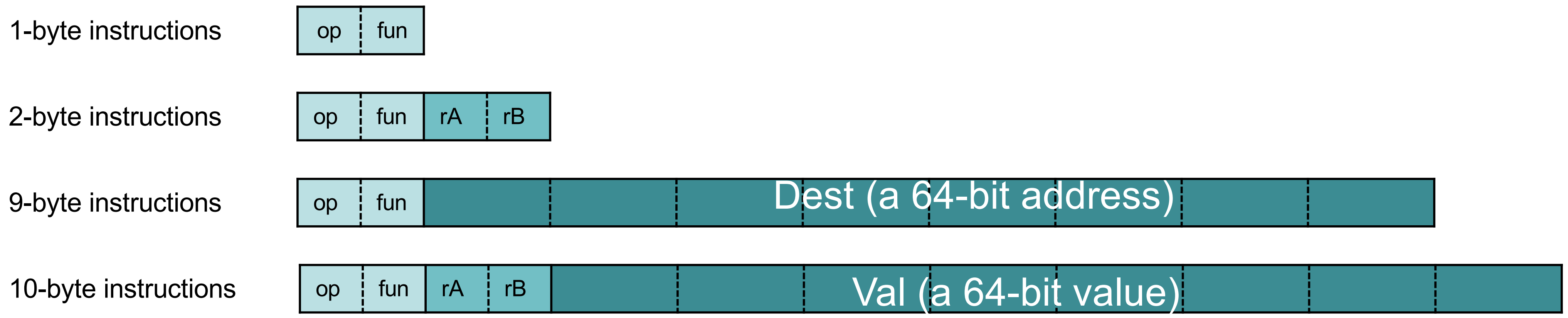
DMEM: Memory

Memory (DMEM)

- You think of this as “main memory” or “How much DRAM my machine has”
- Byte-addressable storage array
- Words stored in **Little Endian** order

Y86 Instruction Classes

- **Simple instructions: `halt, nop`**
- **Move instructions: `rrmovq*, irmovq, rmmovq, mrmovq`**
- Stack operations: `pushq, popq`
- Arithmetic and Logical operations: `addq, subq, andq, xorq`
- Jump instructions: `jmp, jle, jl, je, jne, jg, jge`
- Function call and return instructions: `call, ret`



Simple Instructions

halt	0	0	Set program status register to HLT
nop	1	0	Do nothing