Y86 Introduction

- Topics
  - Move instructions
- Learning Outcomes
  - Explain what each of the MOVE instructions do
  - Explain how addressing works on y86
- Bryant and O'Halloran
  - Section 4.1.2

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

MOVE Instructions

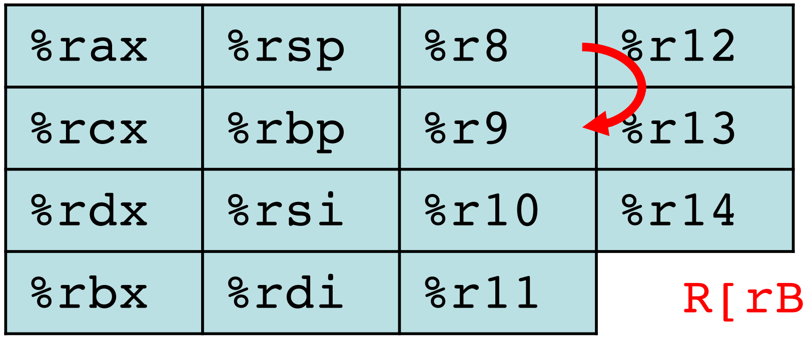
- Move data/numbers to/from registers

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

DMEM:
Memory

MOVE Instructions (Reg->Reg)

- Move data from one register to another



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

R[rB] <- R[rA]

DMEM:
Memory

2-byte instruction

2	0	rA	rB

RRMOVQ %r8, %r9

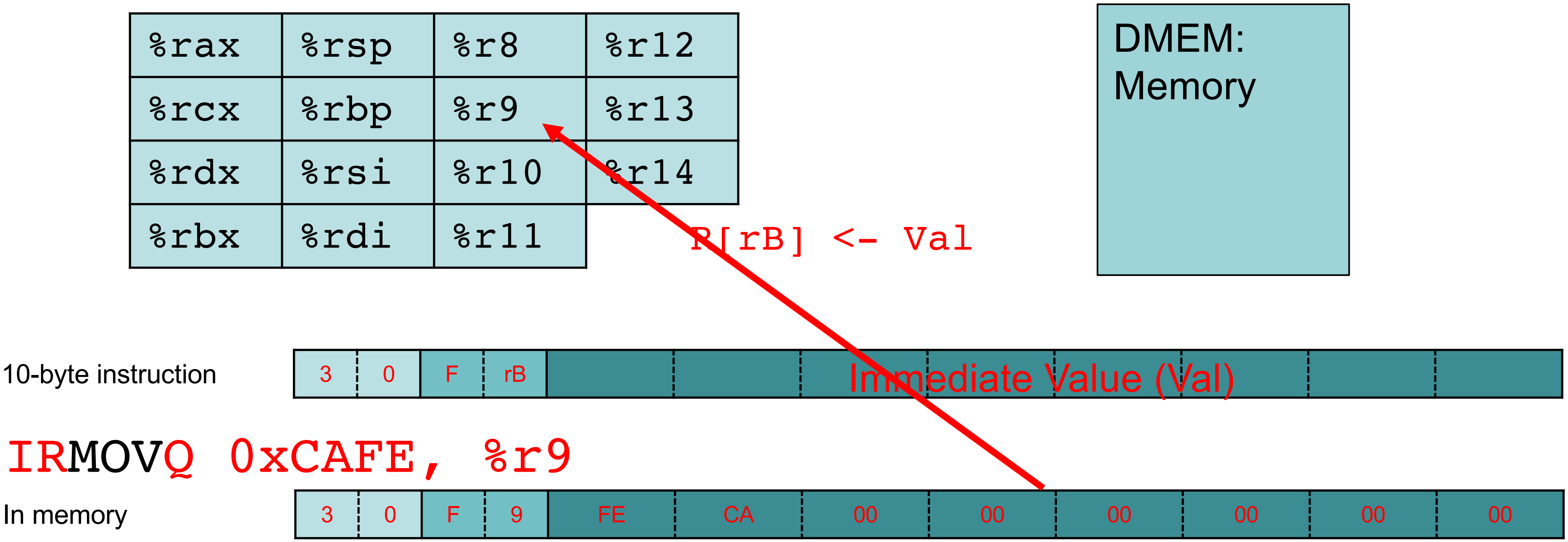
In memory

2	0	8	9

Encoding: 0x20 0x89

MOVE Instructions (Const->Reg)

- Move value (immediate) to a register



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

DMEM: Memory

R[rB] <- Val

10-byte instruction

3	0	F	rB	Immediate Value (Val)						

IRMOVQ 0xCAFE, %r9

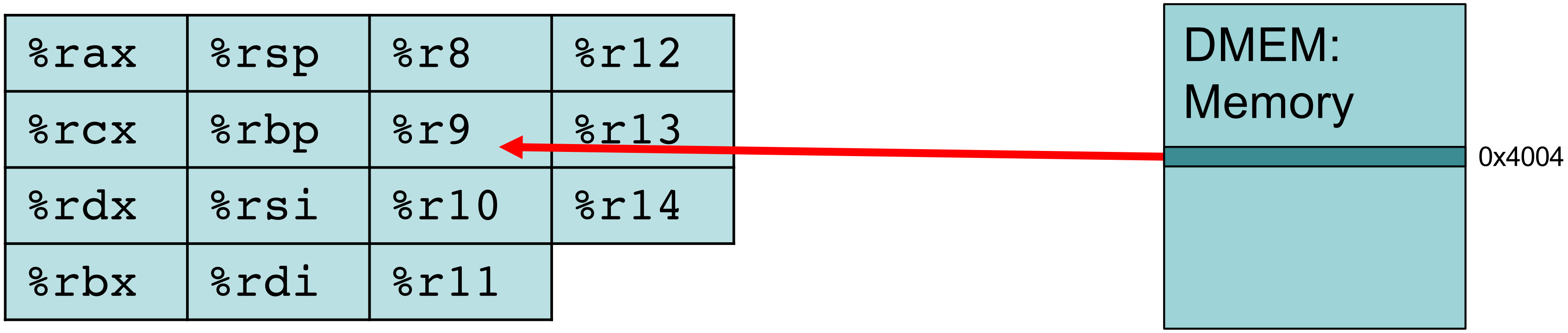
In memory

3	0	F	9	FE	CA	00	00	00	00	00	00

Encoding: 0x30 0xF9 0x000000000000CAFE

MOVE Instructions (Mem->Reg)

- Move value from memory to a register



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

R[rA] <- M_8[D + R[rB]]

10-byte instruction	5	0	rA	rB	Address Offset (D)						

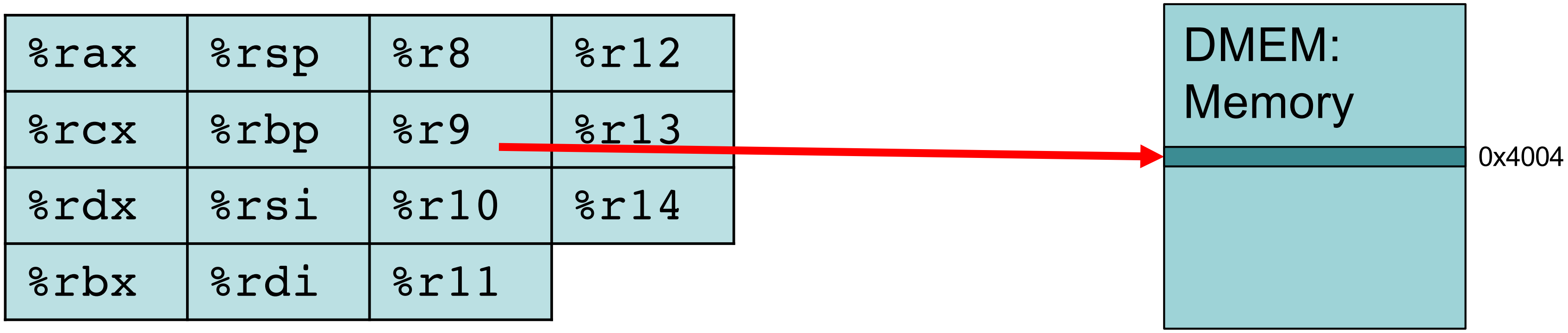
MRMOVQ 4(%rsp), %r9 (Assume R[%rsp] = 0x4000)

In memory	5	0	9	4	04	00	00	00	00	00	00	00

Encoding: 0x50 0x94 0x0000000000000004

MOVE Instructions (Reg->Mem)

- Move value from a register to memory



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

DMEM:
Memory

M_8[D + R[rB]] <- R[rA]

10-byte instruction

4	0	rA	rB	Address Offset							

RMMOVQ %r9, 4(%rsp) (Assume rsp = 0x4000)

In memory

4	0	9	4	04	00	00	00	00	00	00	00

Encoding: 0x40 0x94 0x0000000000000004