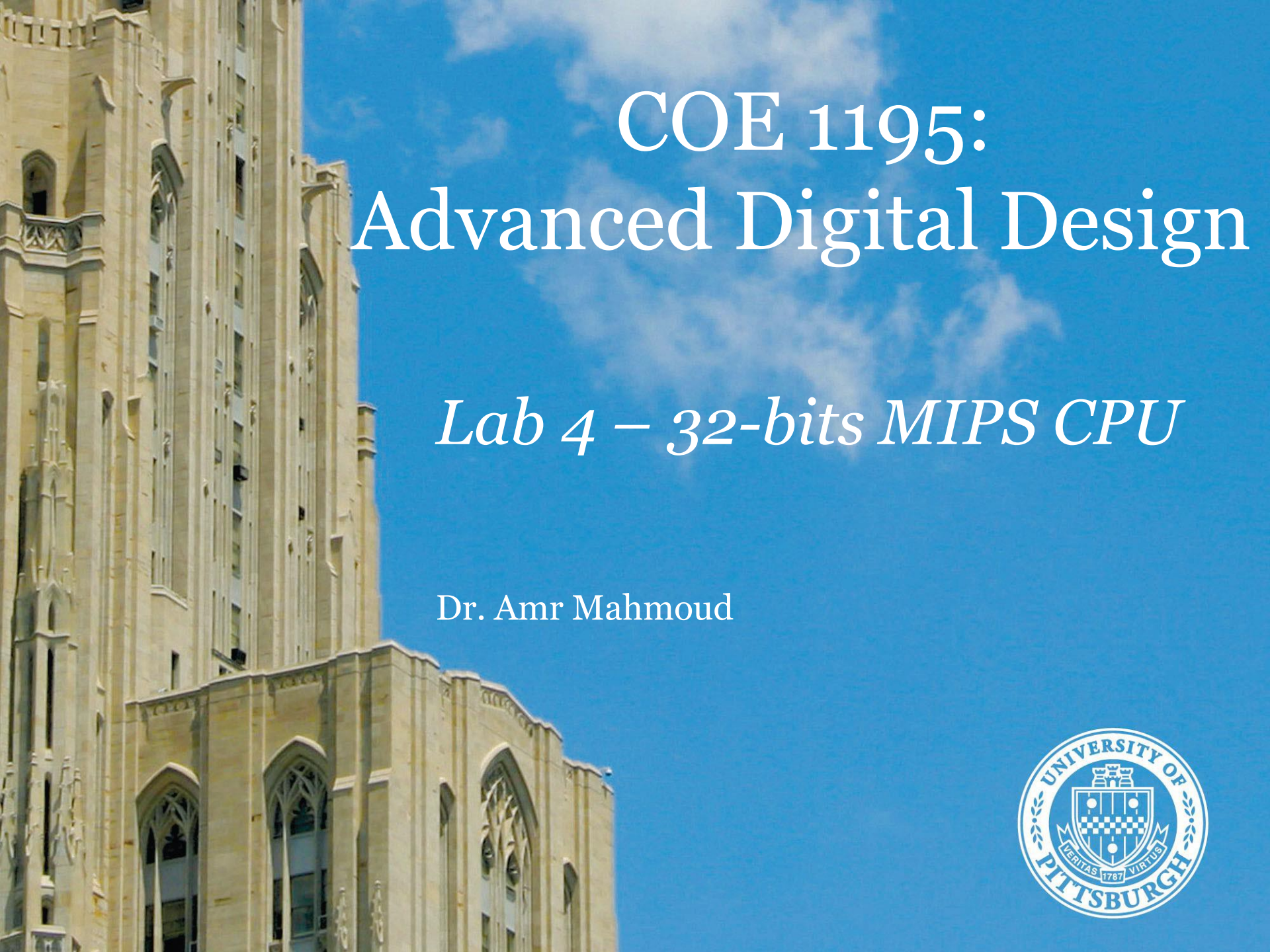
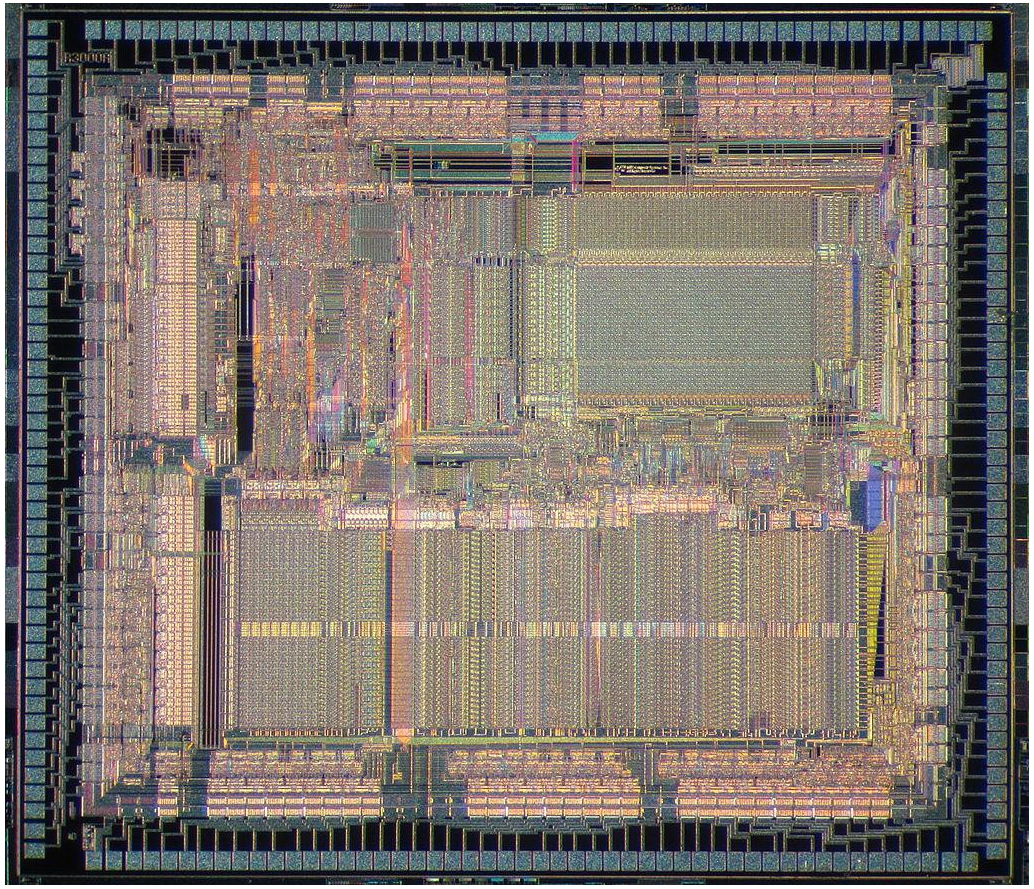


COE 1195: Advanced Digital Design

Lab 4 – 32-bits MIPS CPU

Dr. Amr Mahmoud

MIPS32 Instruction Set Architecture



MIPS R3000 Die photo

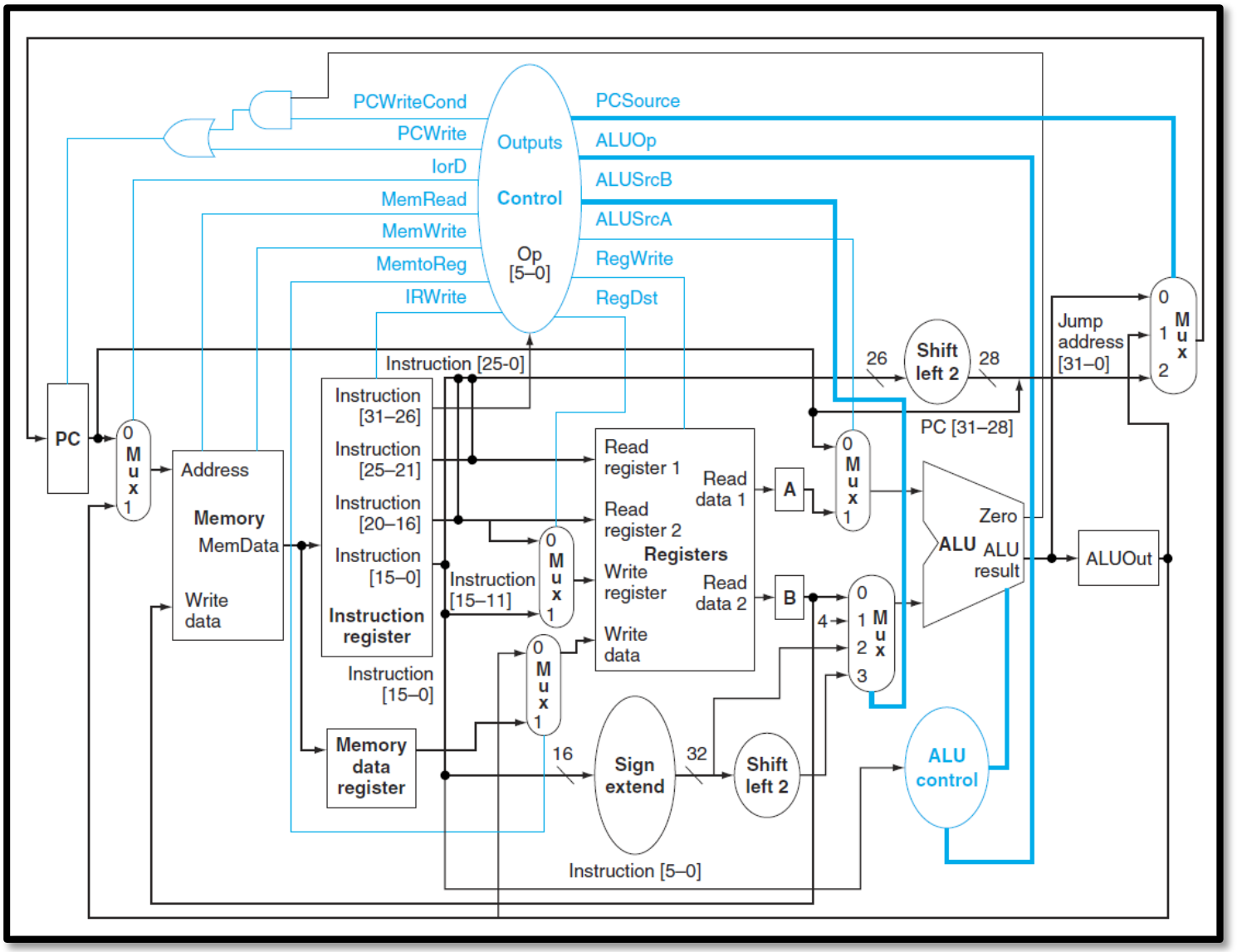
Heavily used in embedded systems and many commercial products (Nintendo, Playstation, countless devices)

RISC Machine Architecture

Load / Store Architecture

32 bit Architecture

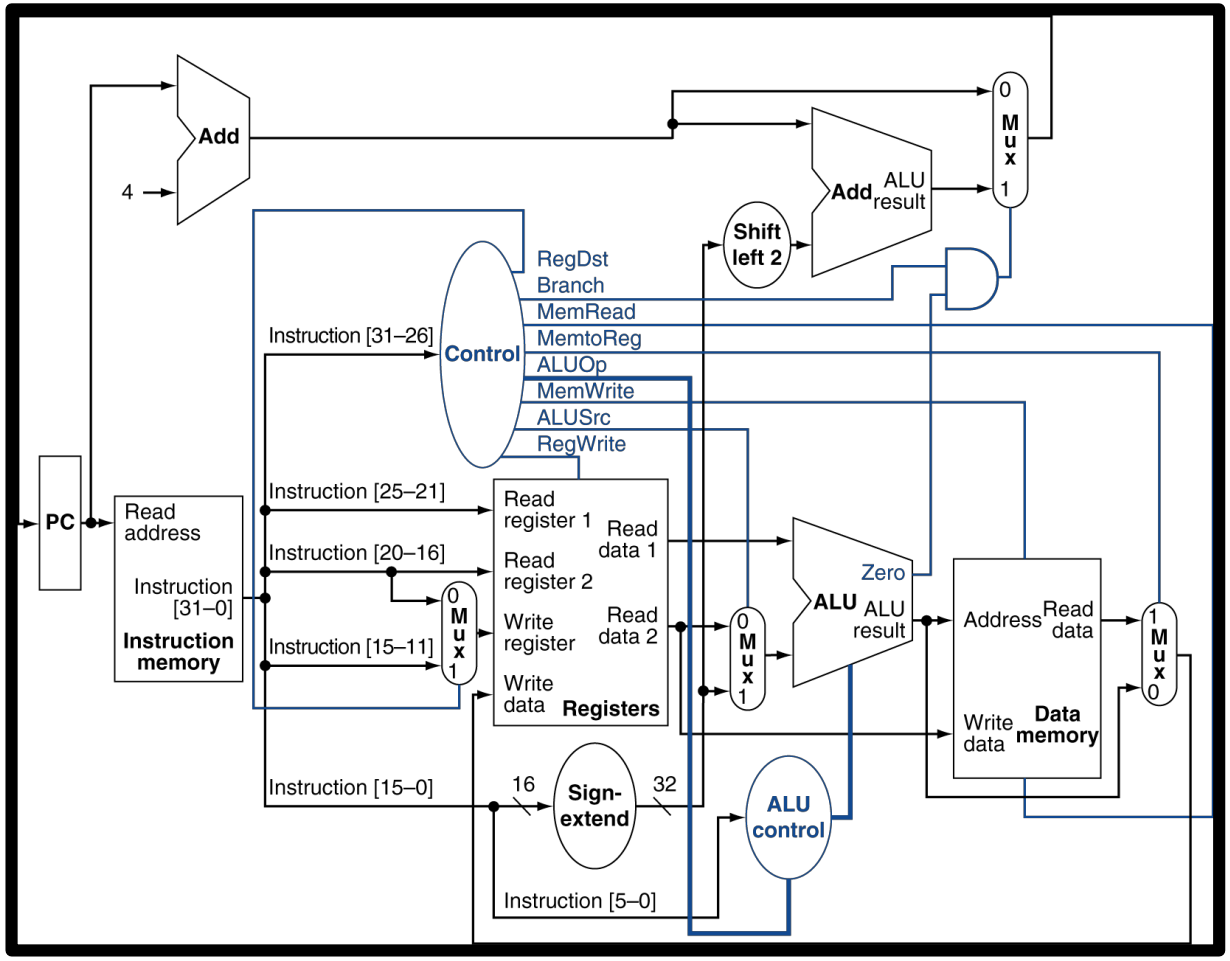
You will implement the MIPS32 ISA



MIPS Multicycle CPU Datapath

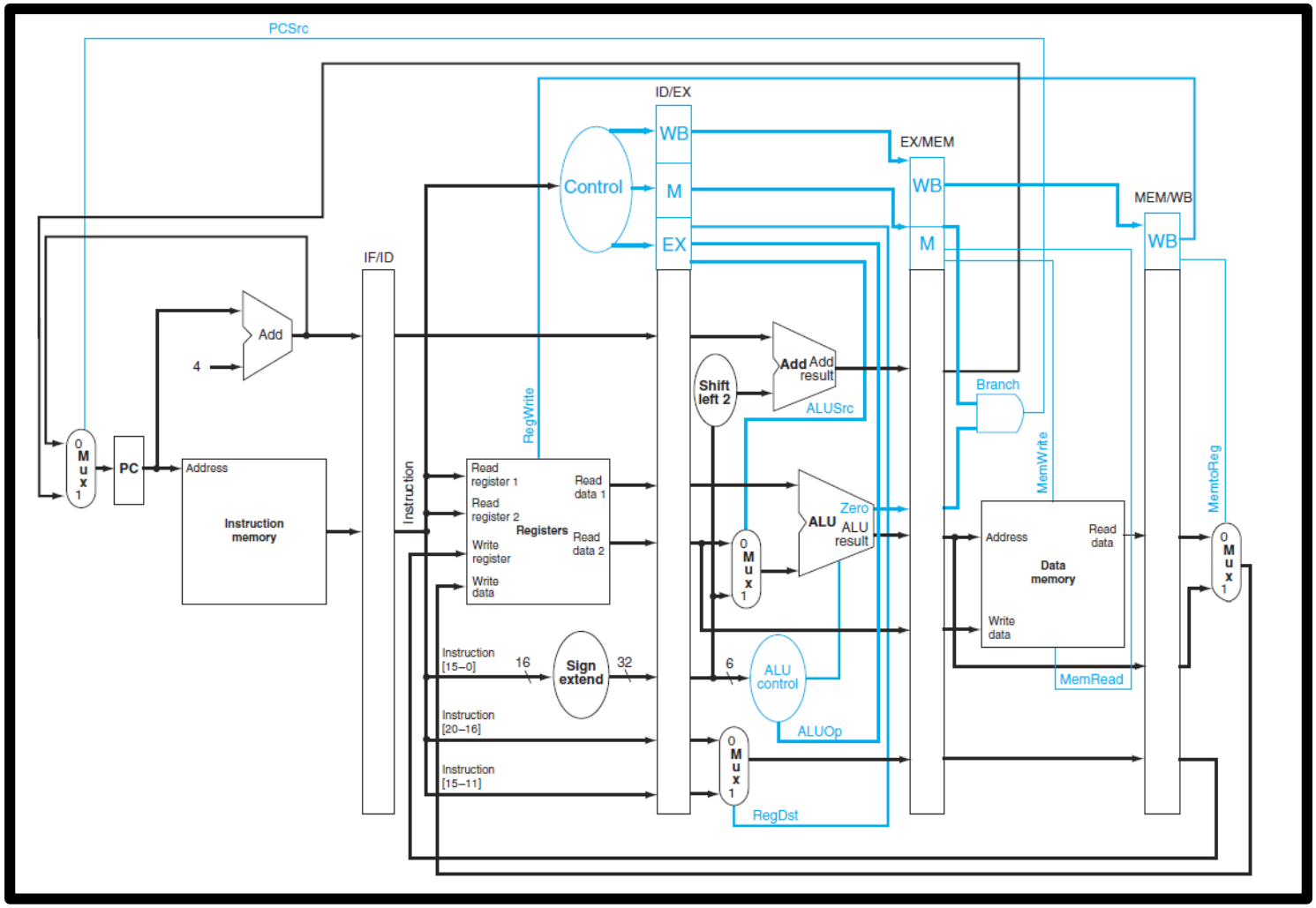
Datapath and Control

MIPS Single Cycle CPU



MIPS Single Cycle CPU Datapath

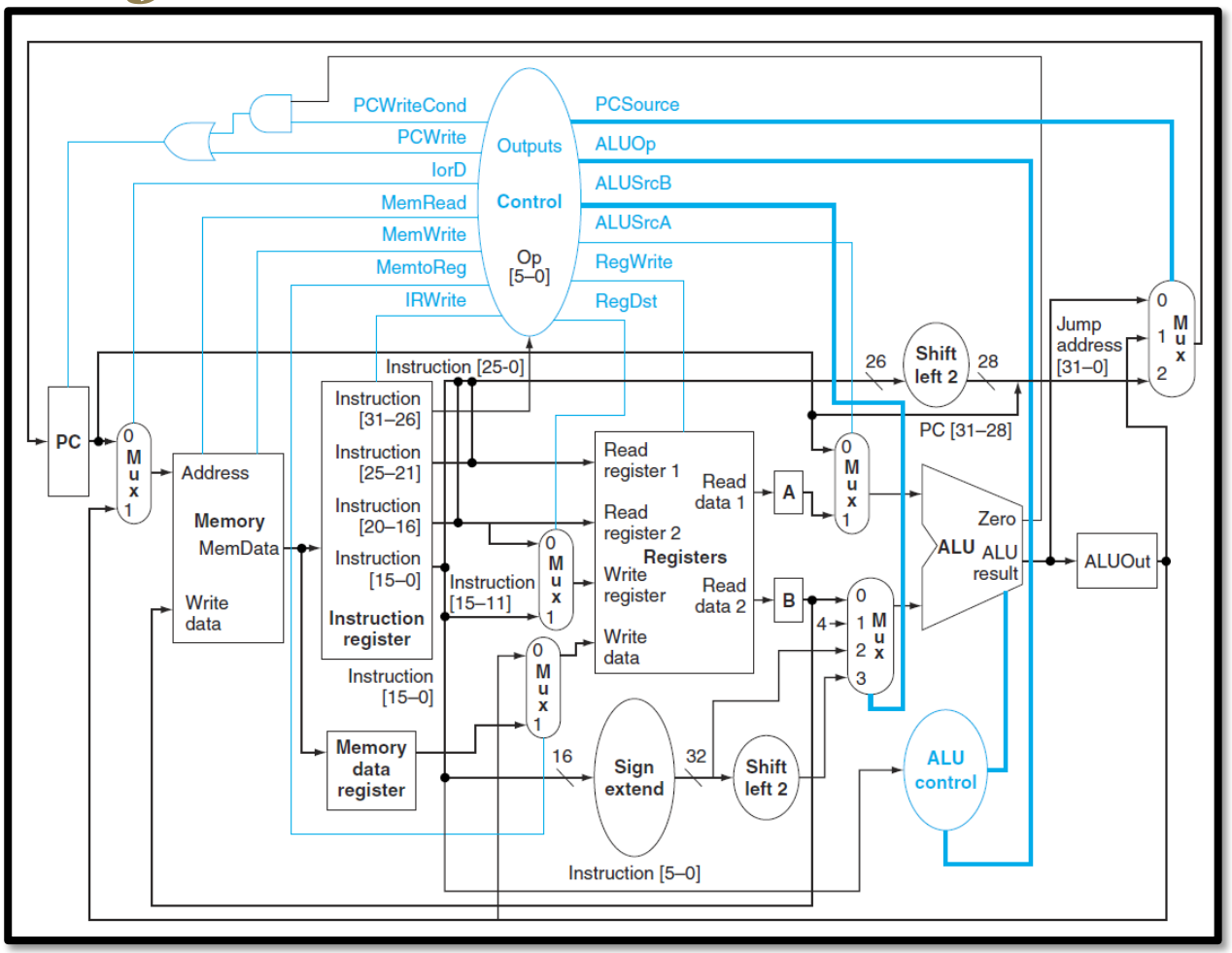
Datapath and Control
Pipeline



MIPS Pipelined CPU Datapath

Datapath and Control

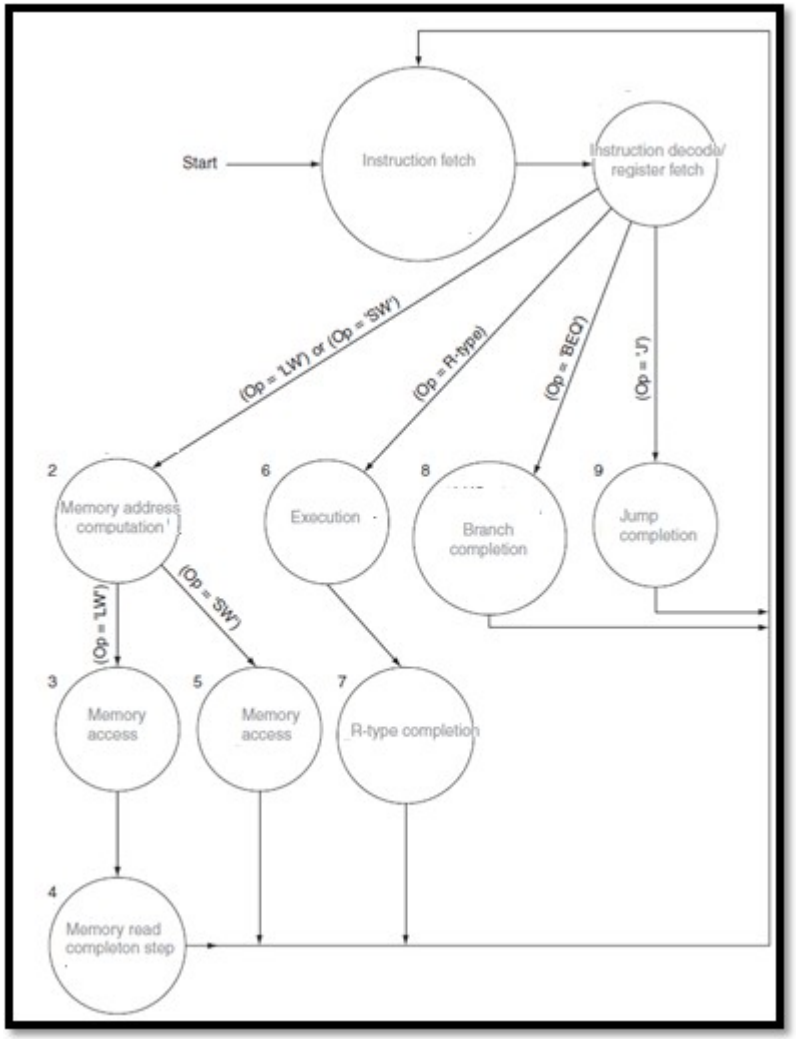
MIPS Multicycle CPU FSM



MIPS Multicycle CPU Datapath

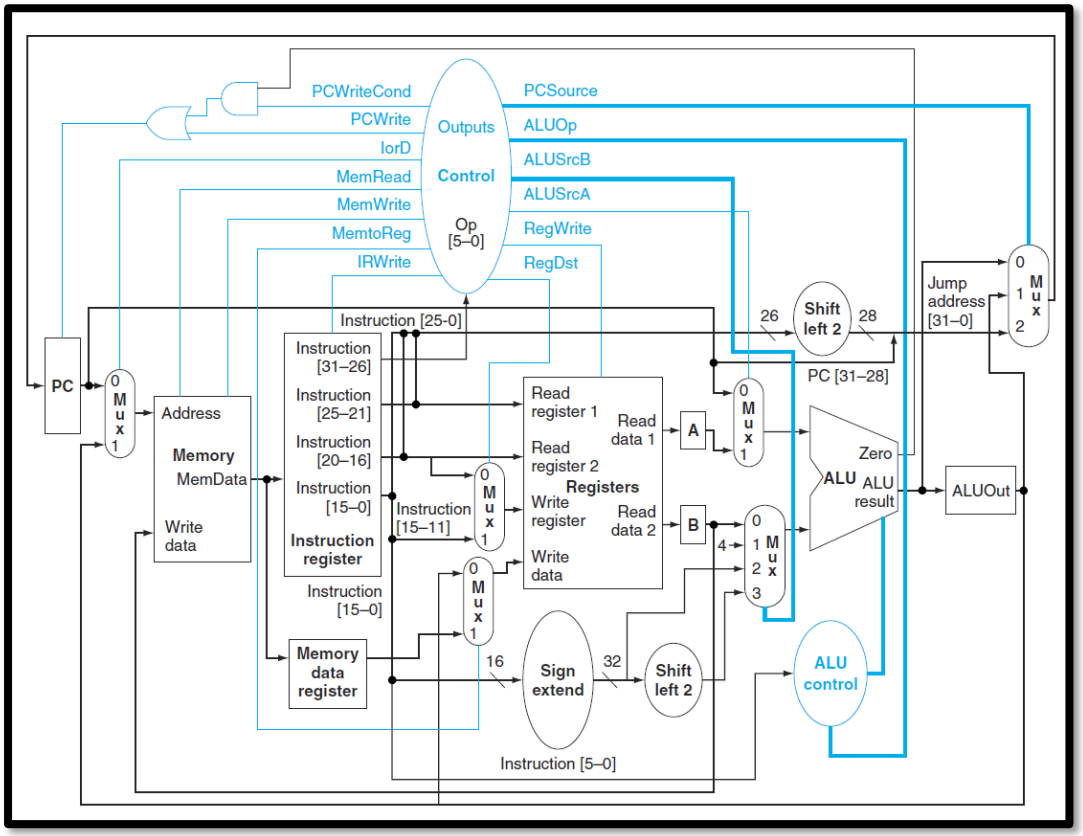
Datapath and Control

MIPS Multicycle CPU FSM

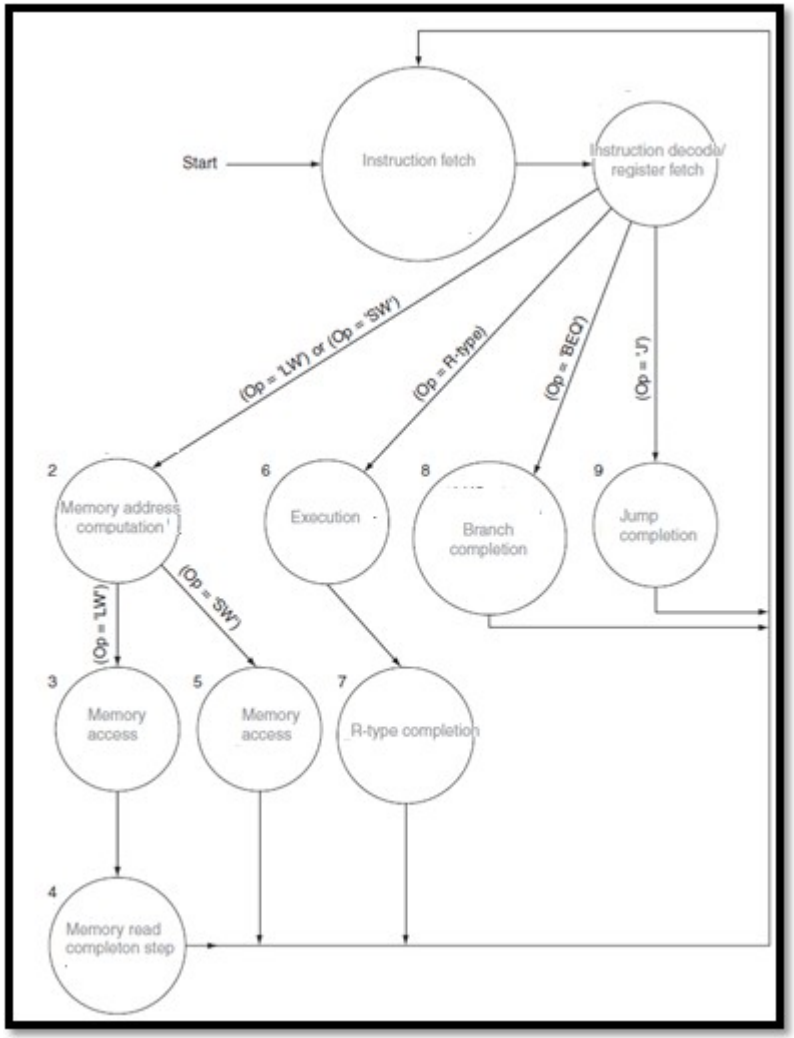


Datapath and Control

MIPS Multicycle CPU FSM



MIPS Multicycle CPU Datapath



MIPS Multicycle CPU FSM Controller

MIPS32 Instruction Set Architecture

MIPS TECHNOLOGIES

MIPS32™ Architecture For Programmers
Volume I: Introduction to the MIPS32™ Architecture

Document Number: MD00082
Revision 0.95
March 12, 2001

MIPS Technologies, Inc.
1225 Charleston Road
Mountain View, CA 94043-1353

Table 4-23 describes the fields used in these instructions.

Table 4-23 CPU Instruction Format Fields

Field	Description
opcode	6-bit primary operation code
rd	5-bit specifier for the destination register
rs	5-bit specifier for the source register
rt	5-bit specifier for the target (source/destination) register or used to specify functions within the primary *opcode* REGIMM
immediate	16-bit signed *immediate* used for logical operands, arithmetic signed operands, load/store address byte offsets, and PC-relative branch signed instruction displacement
instr_index	26-bit index shifted left two bits to supply the low-order 28 bits of the jump target address
sa	5-bit shift amount
function	6-bit function field used to specify functions within the primary *opcode* SPECIAL

Figure 4-1 Immediate (I-Type) CPU Instruction Format

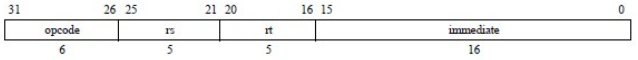


Figure 4-2 Jump (J-Type) CPU Instruction Format

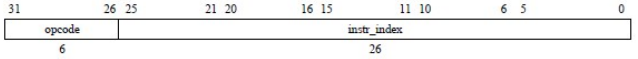
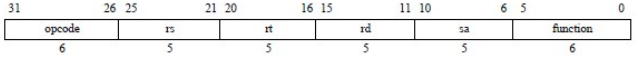


Figure 4-3 Register (R-Type) CPU Instruction Format



MIPS32 Instruction Set Architecture

Add Unsigned Word — **ADDU**

31–26	25–21	20–16	15–11	10–6	5–0
SPECIAL 000000	rs	rt	rd	0 00000	ADDU 100001
6	5	5	5	5	6

Format: `ADDU rd, rs, rt` — MIPS32 (MIPS I)

Purpose:

To add 32-bit integers

Description: `rd ← rs + rt`

The 32-bit word value in GPR *rt* is added to the 32-bit value in GPR *rs* and the 32-bit arithmetic result is placed into GPR *rd*.

No Integer Overflow exception occurs under any circumstances.

Restrictions:

None

Operation:

```
temp ← GPR[rs] + GPR[rt]
GPR[rd] ← temp
```

Exceptions:

None

Programming Notes:

The term "unsigned" in the instruction name is a misnomer; this operation is 32-bit modulo arithmetic that does not trap on overflow. This instruction is appropriate for unsigned arithmetic, such as address arithmetic, or integer arithmetic environments that ignore overflow, such as C language arithmetic.