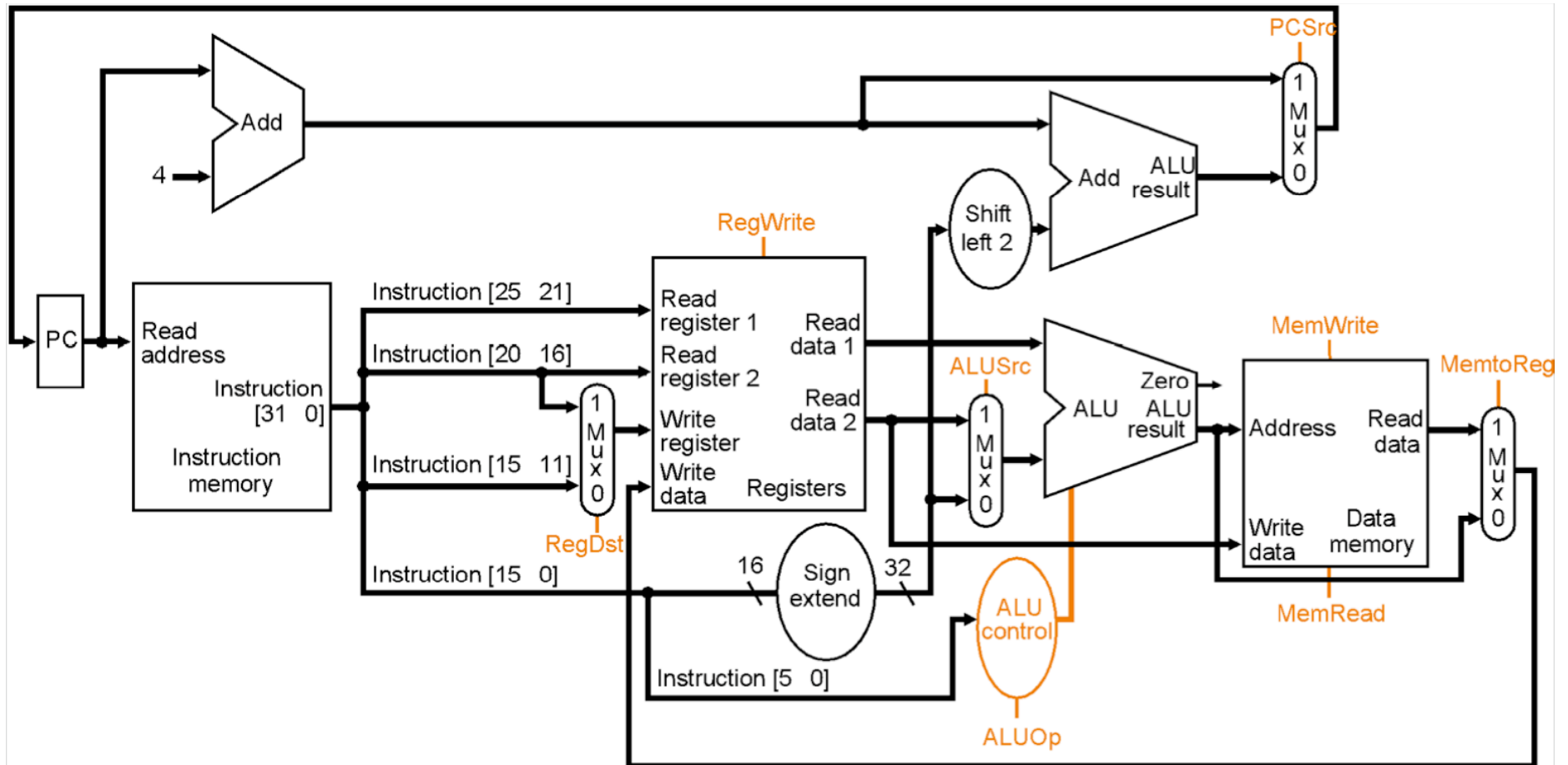


Today

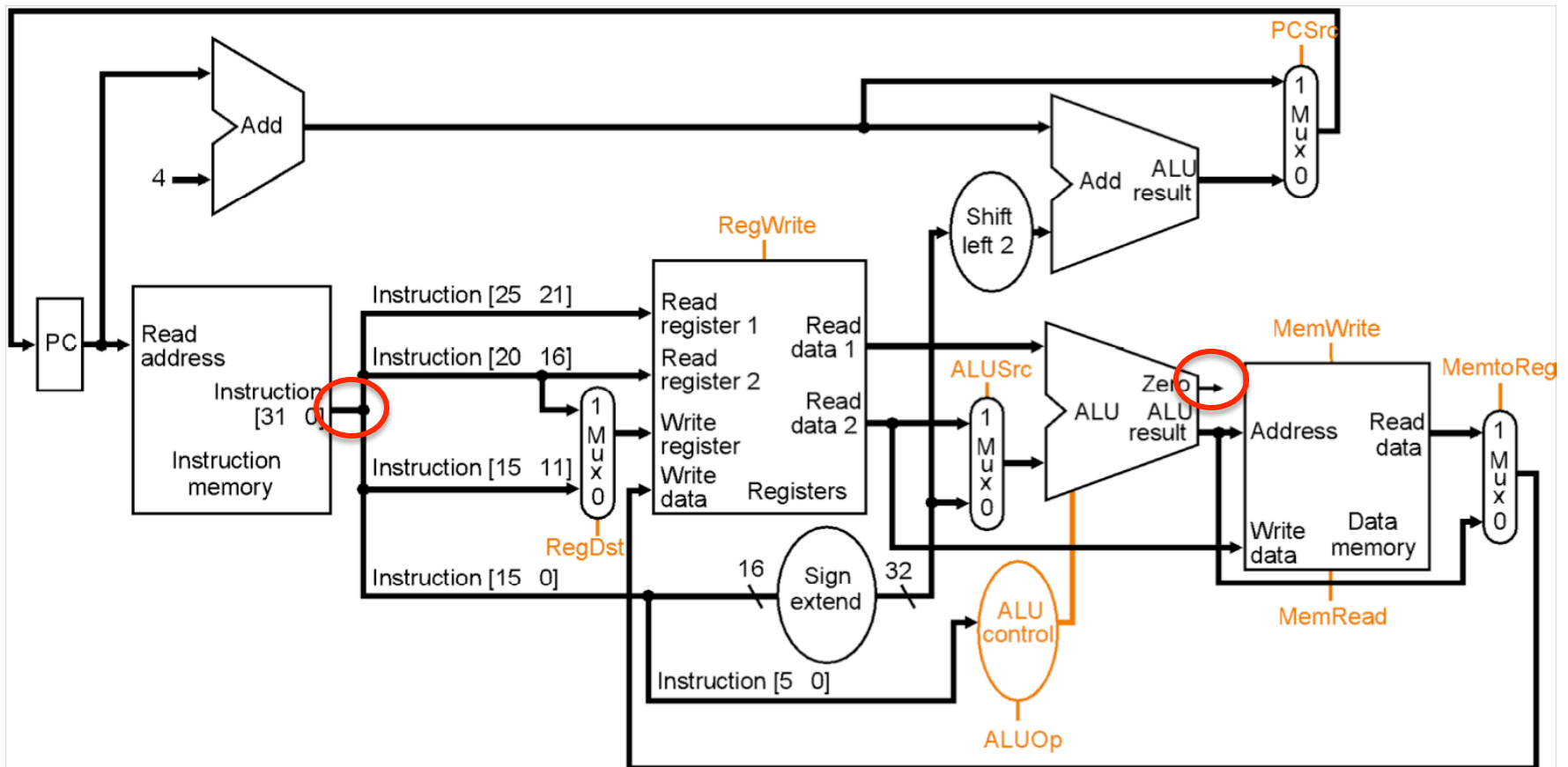
- Project extension!
- Finish single-cycle processor design
- Maybe discuss pipelining
- Switch statements in x86
- Hello world shoot-out
- Hand back a bunch of stuff

The complete datapath (without jumps)



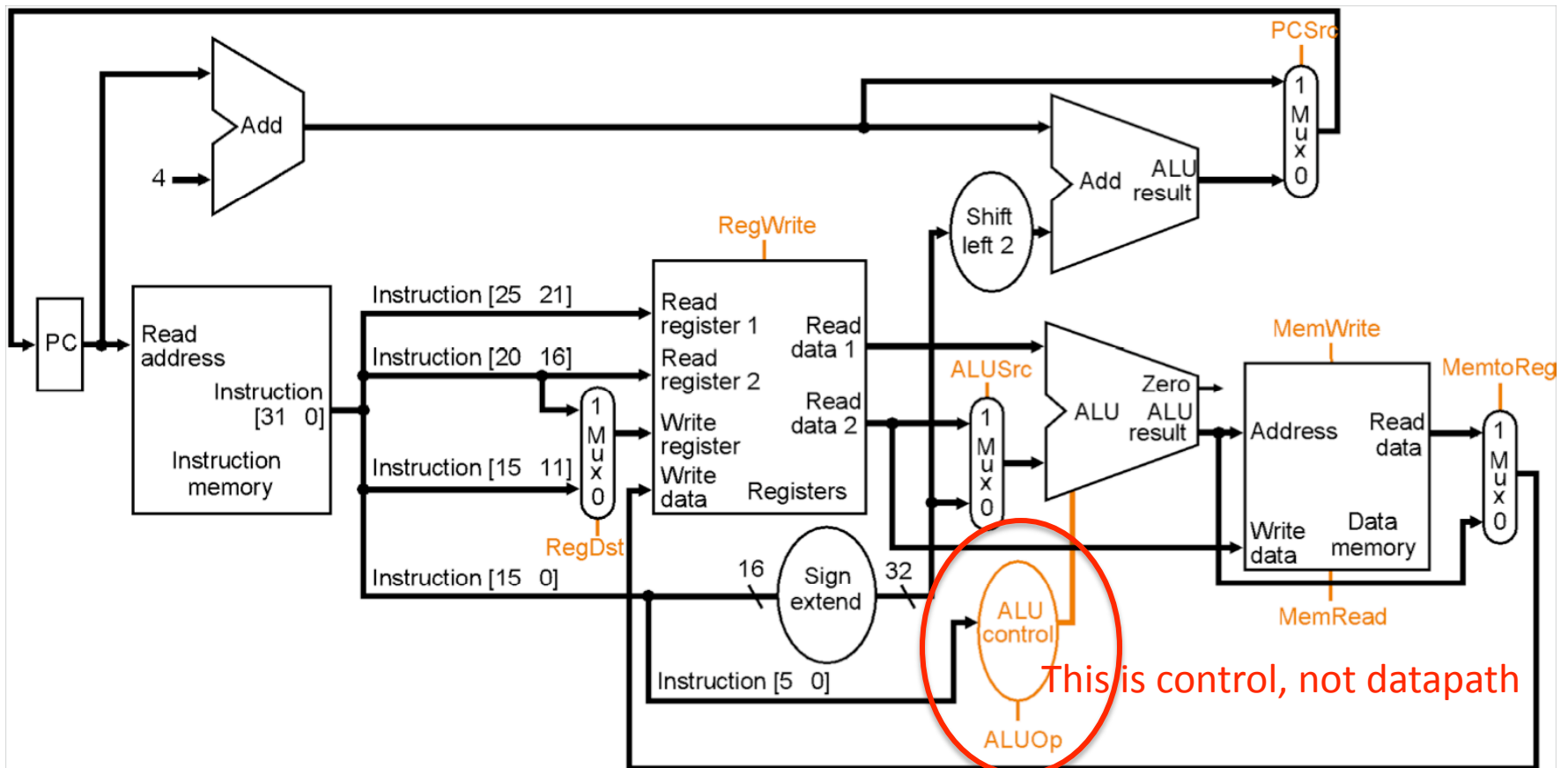
Control Signals (Datapath -> Control)

- Instruction opcode (`opc`)
- Instruction func (`func`)
- Branch outcome (`zero` means `rt == rs`)



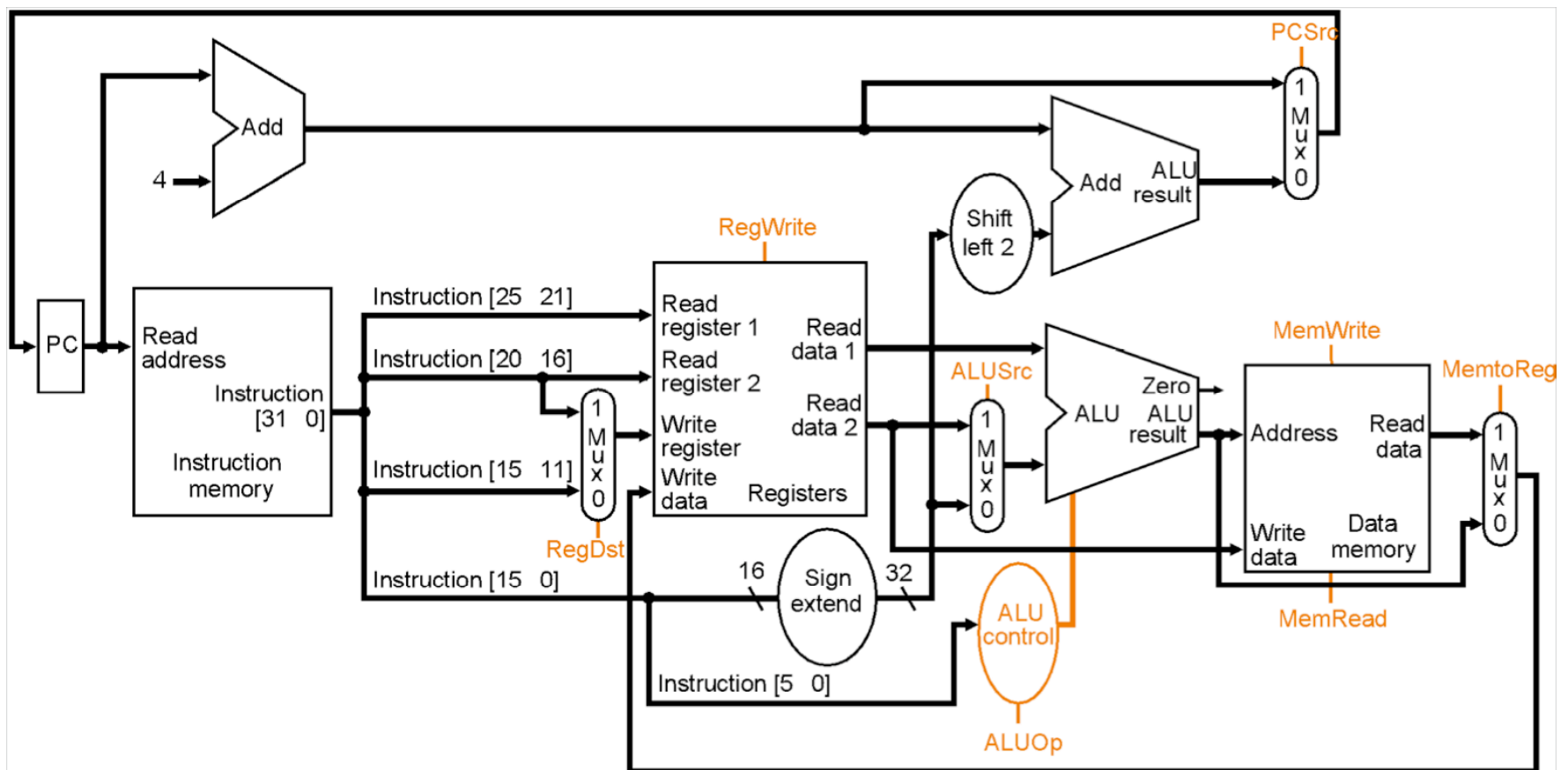
Control Signals (Control -> Datapath)

Signal	== 0	== 1
RegDst	Write to rd	Write to rt
RegWrite	Register writes suppressed	Register writes occur
ALUSrc	2 nd ALU input is R[rd]	2 nd ALU input is the immediate
ALUOp	Multiple bits; value determines the operation the ALU will perform.	



Control Signals (Control -> Datapath)

Signal	== 0	== 1
PCSrc	$PC \leq PC + 4$	$PC \leq PC + 4 + \text{immediate}$
MemRead	Do not read data memory	Perform read at address
MemWrite	Do not write data memory	Perform write at address
MemtoReg	Present ALU result to register file for write	Present ALU result to register file for write.



Computing Control Signals

```
opc    = Inst[31:26];    func    = Inst[31:26];
rs     = Inst[25:21];    rt     = Inst[20:16];
rd     = Inst[15:11];    sa     = Inst[10:6];
imm    = Inst[15:0];

WritesRD = (opc == `OP_RRR)
WritesRT = ( (opc == `OP_ADDI) || (opc == `OP_ADDIU)
             || (opc == `OP_ANDI) || (opc == `OP_LBU)
             ...);

IsStore = ( (opc == `OP_SB)   || (opc == `OP_SH)
            || (opc == `OP_SW));

IsLoad  = ( (opc == `OP_LBU)  || (opc == `OP_LHU)
            || (opc == `OP_LW));

IsBranch = ( (opc == `OP_BEQ)  || (opc == `OP_BNE)
             || (opc == `OP_BLEZ) || (opc == `OP_BGTZ)
             || ..
             )

ReadsRT = ( (opc == `OP_RRR)  || (opc == `OP_BEQ)
            || (opc == `OP_BNE) || (IsStore);
```

Computing Control Signals

```
RegDst    = WritesRD;
```

```
RegWrite  = (WritesRD | WritesRT);
```

```
MemWrite  = IsStore;
```

```
MemRead   = IsLoad;
```

```
MemToReg  = IsLoad;
```

```
ALUSrc    = ReadsRT & ~IsStore;
```

```
    // careful - some diagrams seem
```

```
    // inconsistent about ALUSrc "mux sense"
```

```
PCSrc     = IsBranch && BranchTaken; //from ALU
```

```
ALUOp = // conversion between opc/func and muxes  
        // inside ALU. A function of opc and func
```

A Single-cycle Processor

- Performance refresher

$$ET = IC * CPI * CT$$

- Single cycle \Rightarrow $CPI == 1$; That sounds great!
- Unfortunately, Single cycle \Rightarrow CT is large
 - Even RISC instructions take quite a bite of effort to execute
 - This is a lot to do in one cycle