MIPS Datapath – Single Memory – No Pipelining

Prof. James L. Frankel
Harvard University

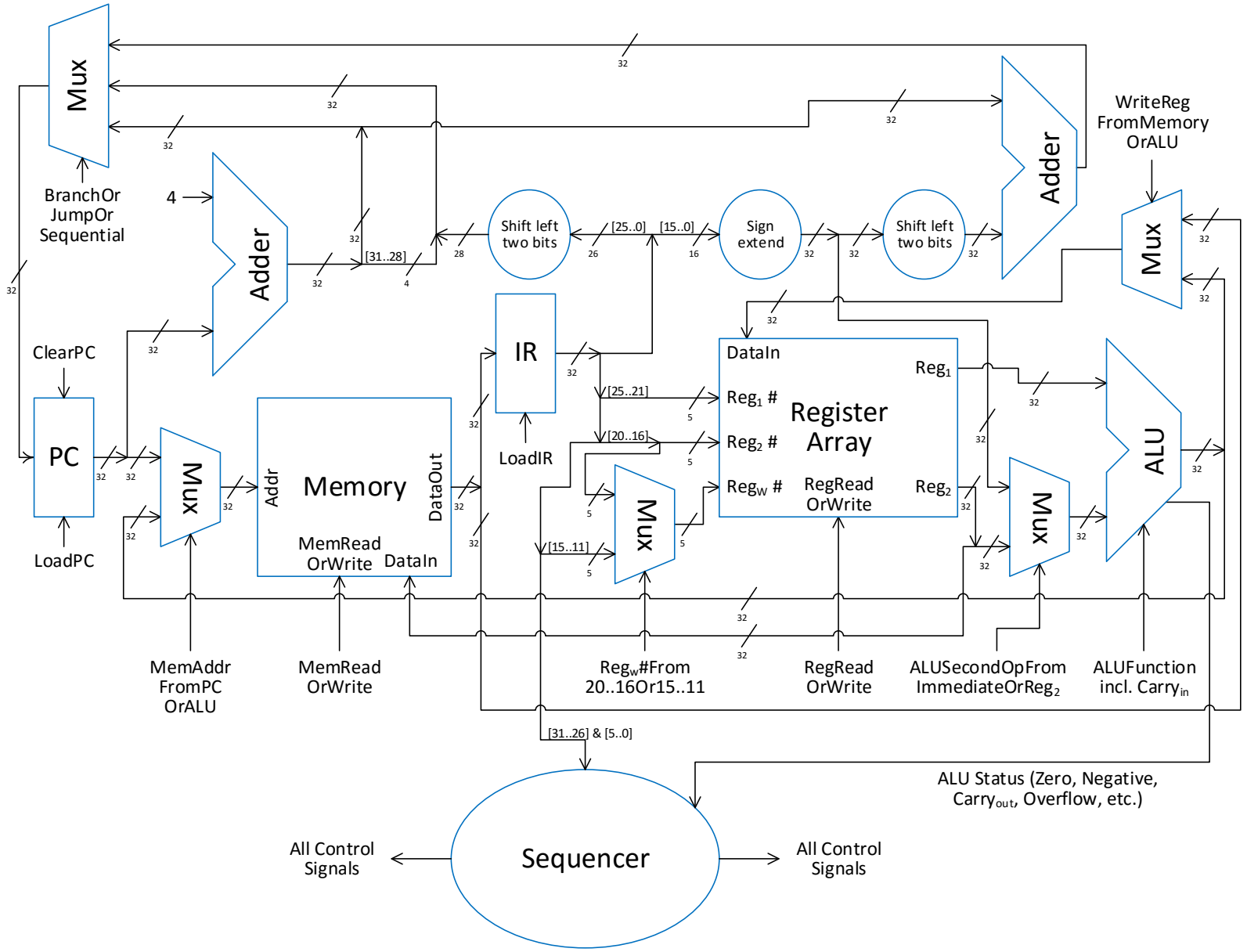
Version of 5:28 PM 23-Feb-2016

32-bit MIPS Datapath

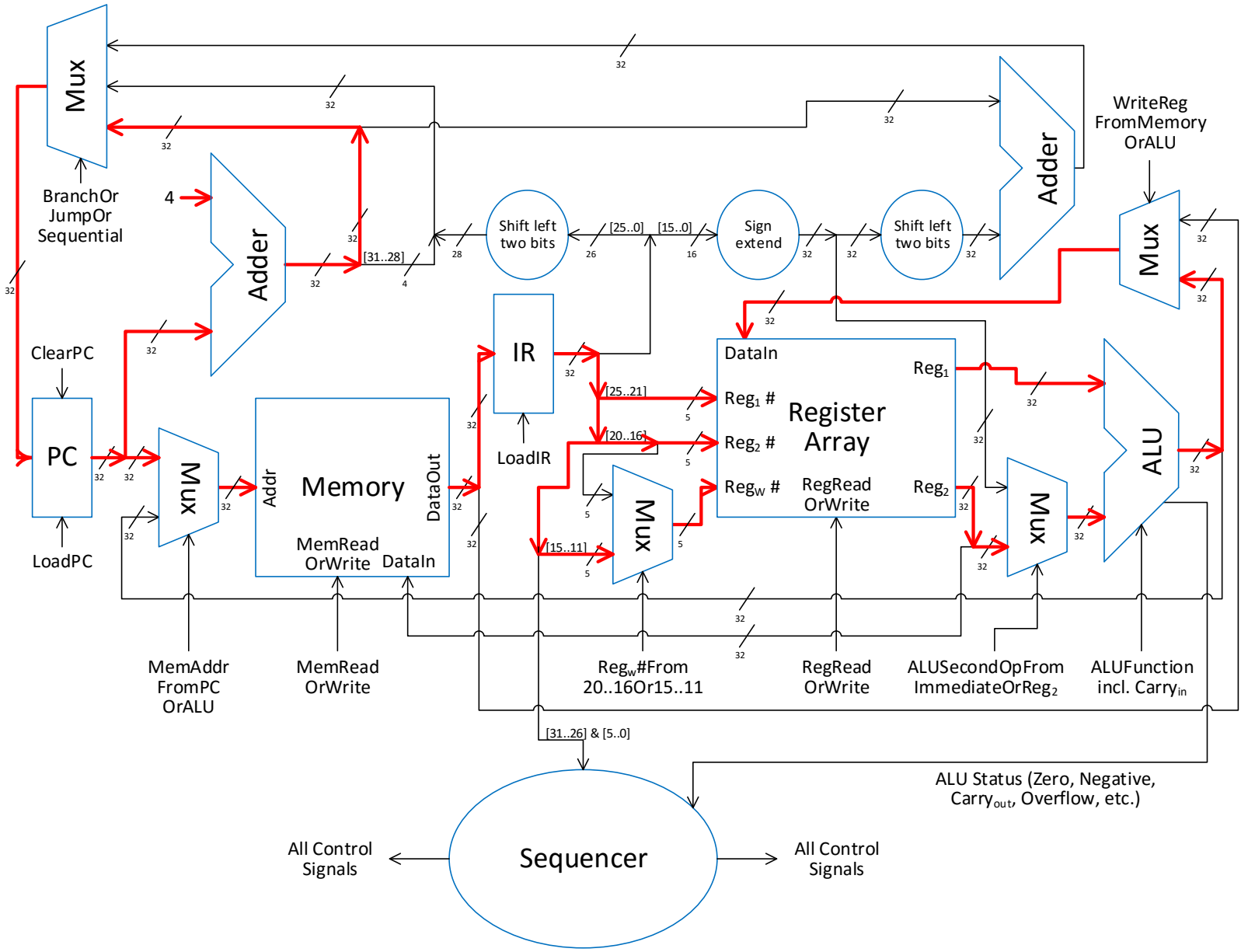
- One memory
  - Instruction memory & data memory are combined in a single memory
  - No pipelining
    - Multicycle

Notes on the Block Diagram

- Wires are not directional
  - Arrowheads are present to indicate which component is driving the wire
- Subscripting operator indicates bit selection
  - For example, [25.21] indicates that bits 25 to 21 are selected


Mux
BranchOr
JumpOr
Sequential
4
Adder
[31..28]
Shift left
two bits
[25..0]
[15..0]
Sign
extend
Shift left
two bits
Adder
WriteReg
FromMemory
OrALU
Mux
ClearPC
PC
LoadPC
Mux
Addr
Memory
DataOut
MemRead
OrWrite
DataIn
IR
LoadIR
[25..21]
[20..16]
[15..11]
Mux
DataIn
Reg1
Reg1 #
Reg2 #
Regw #
Register
Array
RegRead
OrWrite
Reg2
Mux
ALU
MemAddr
FromPC
OrALU
MemRead
OrWrite
Regw#From
20..16Or15..11
RegRead
OrWrite
ALUSecondOpFrom
ImmediateOrReg2
ALUFunction
incl. Carryin
[31..26] & [5..0]
ALU Status (Zero, Negative,
Carryout, Overflow, etc.)
All Control
Signals
Sequencer
All Control
Signals

Three-Operand R-Type Instructions


Mux
BranchOr
JumpOr
Sequential
4
Adder
[31..28]
Shift left
two bits
[25..0]
[15..0]
Sign
extend
Shift left
two bits
Adder
WriteReg
FromMemory
OrALU
Mux
ClearPC
PC
LoadPC
Mux
Addr
Memory
DataOut
MemRead
OrWrite
DataIn
IR
LoadIR
[25..21]
[20..16]
[15..11]
Mux
DataIn
Reg1 #
Reg2 #
Regw #
Register
Array
RegRead
OrWrite
Reg1
Reg2
Mux
ALU
MemAddr
FromPC
OrALU
MemRead
OrWrite
Regw#From
20..16Or15..11
RegRead
OrWrite
ALUSecondOpFrom
ImmediateOrReg2
ALUFunction
incl. Carryin
[31..26] & [5..0]
ALU Status (Zero, Negative,
Carryout, Overflow, etc.)
All Control
Signals
Sequencer
All Control
Signals

Three-Operand R-Type Instruction Observations

- Implementation of the SLT (Set on Less Than) and SLTU (Set on Less Than Unsigned) instructions
  - An unusual ALU output would need to be created for these instructions
    - A 32-bit 0- or 1-valued result – not a bit-wise result
    - 1 if true, 0 if false

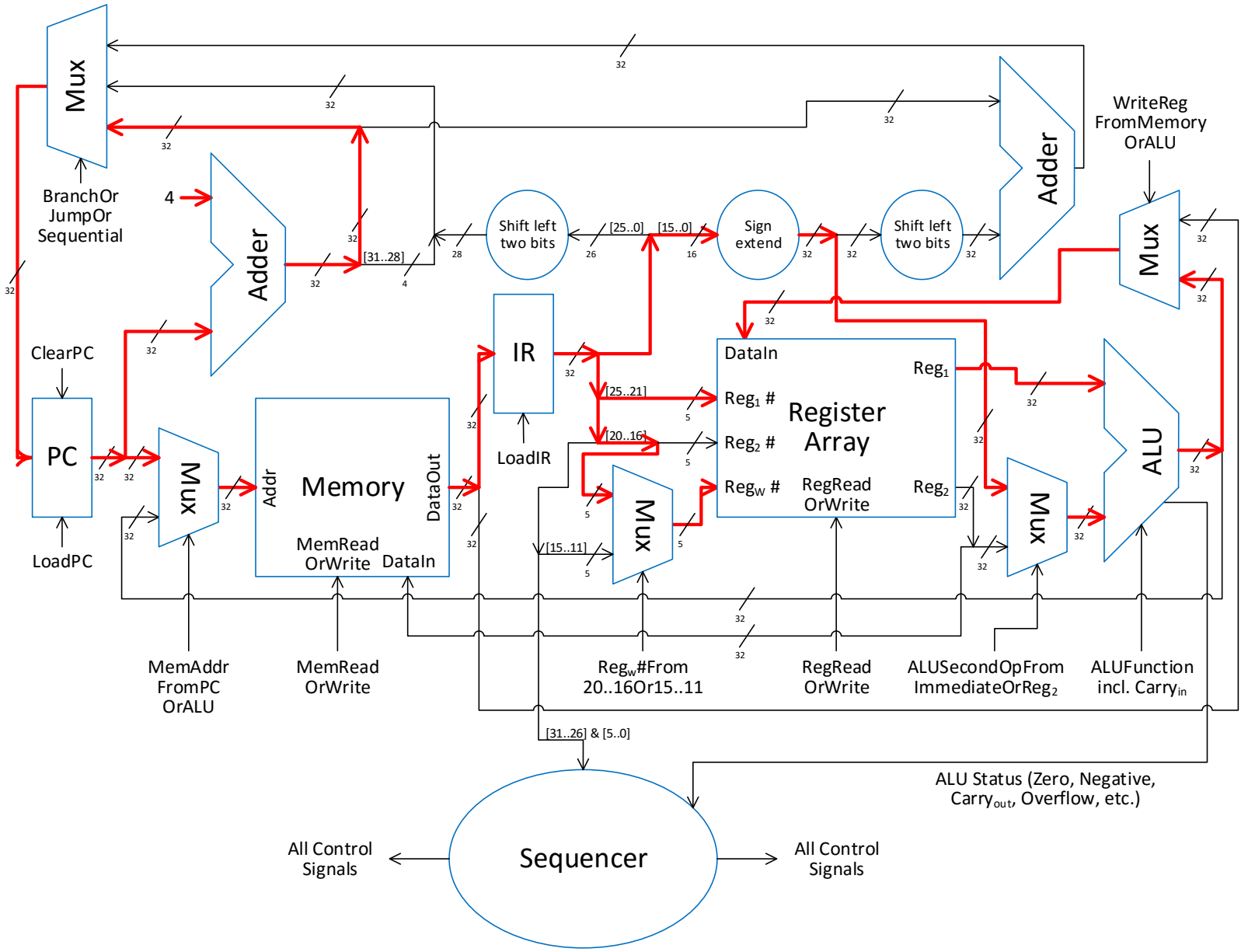
Multiply/Divide R-Type Instructions

- Multiply and Divide instructions are not supported by our block diagram
- Missing components
  - HI and LO registers
  - Multiply and divide hardware
  - Data paths for HI, LO, multiply, and divide hardware

Shift R-Type Instructions

- Shift R-Type instructions are not supported by our block diagram
- Missing components
  - Shifter hardware – possibly a *barrel shifter*
  - Data paths for shifter hardware
    - Including path for the *sa* bits and for the variable length shift distance

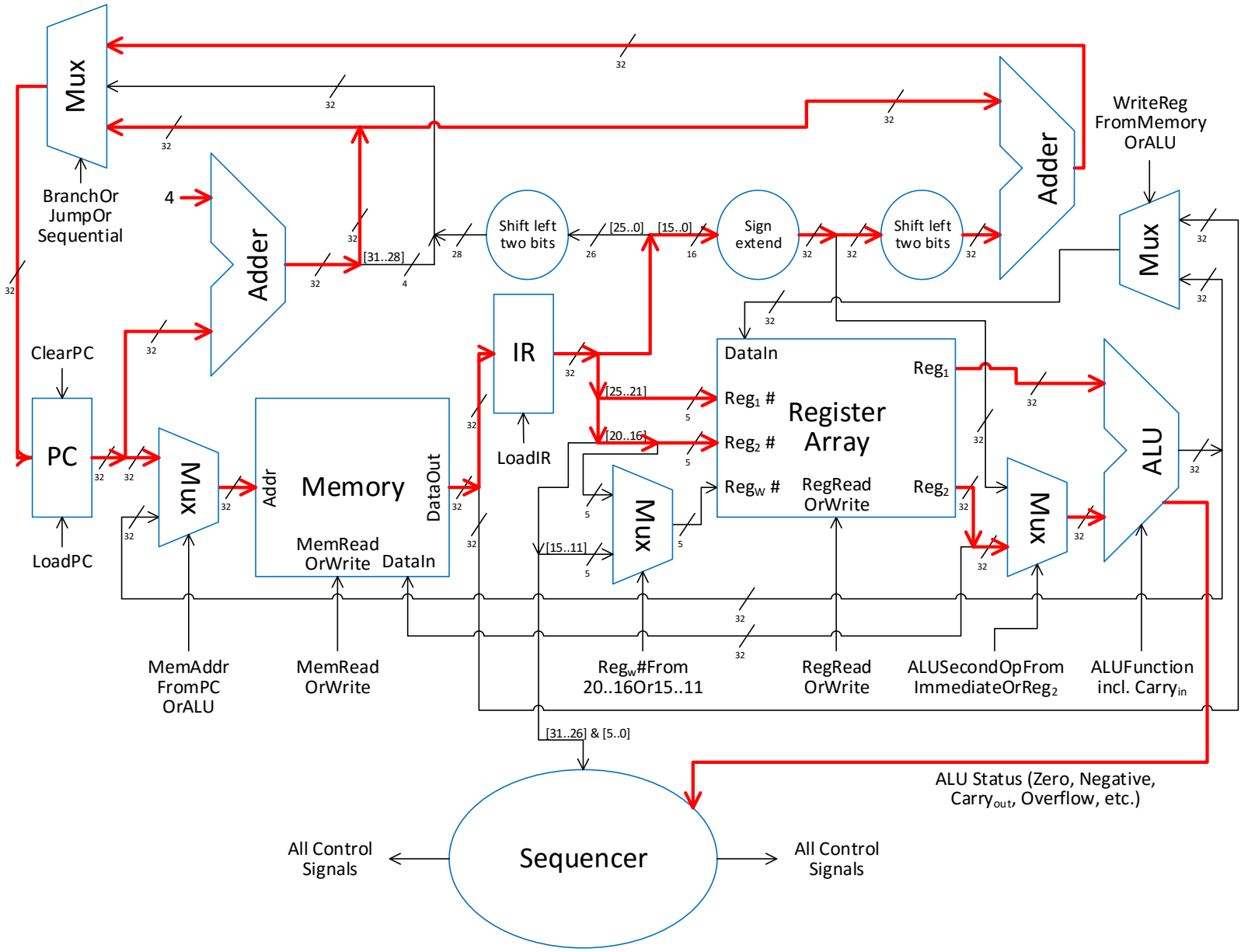
ALU I-Type Instructions


Mux
BranchOr
JumpOr
Sequential
4
Adder
[31..28]
Shift left
two bits
[25..0]
[15..0]
Sign
extend
Shift left
two bits
Adder
WriteReg
FromMemory
OrALU
Mux
ClearPC
PC
LoadPC
Mux
Addr
Memory
DataOut
MemRead
OrWrite
DataIn
IR
LoadIR
[25..21]
[20..16]
[15..11]
Mux
DataIn
Reg1 #
Reg2 #
Regw #
Register
Array
RegRead
OrWrite
Reg1
Reg2
Mux
ALU
MemAddr
FromPC
OrALU
MemRead
OrWrite
Regw#From
20..16Or15..11
RegRead
OrWrite
ALUSecondOpFrom
ImmediateOrReg2
ALUFunction
incl. Carryin
[31..26] & [5..0]
ALU Status (Zero, Negative,
Carryout, Overflow, etc.)
All Control
Signals
Sequencer
All Control
Signals

ALU I-Type Instruction Observations

- The block diagram has data paths to sign extend the immediate operand, but ANDI, ORI, and XORI zero extend the immediate operand
- There is no data path for the implementation of the LUI (Load Upper Immediate) instruction

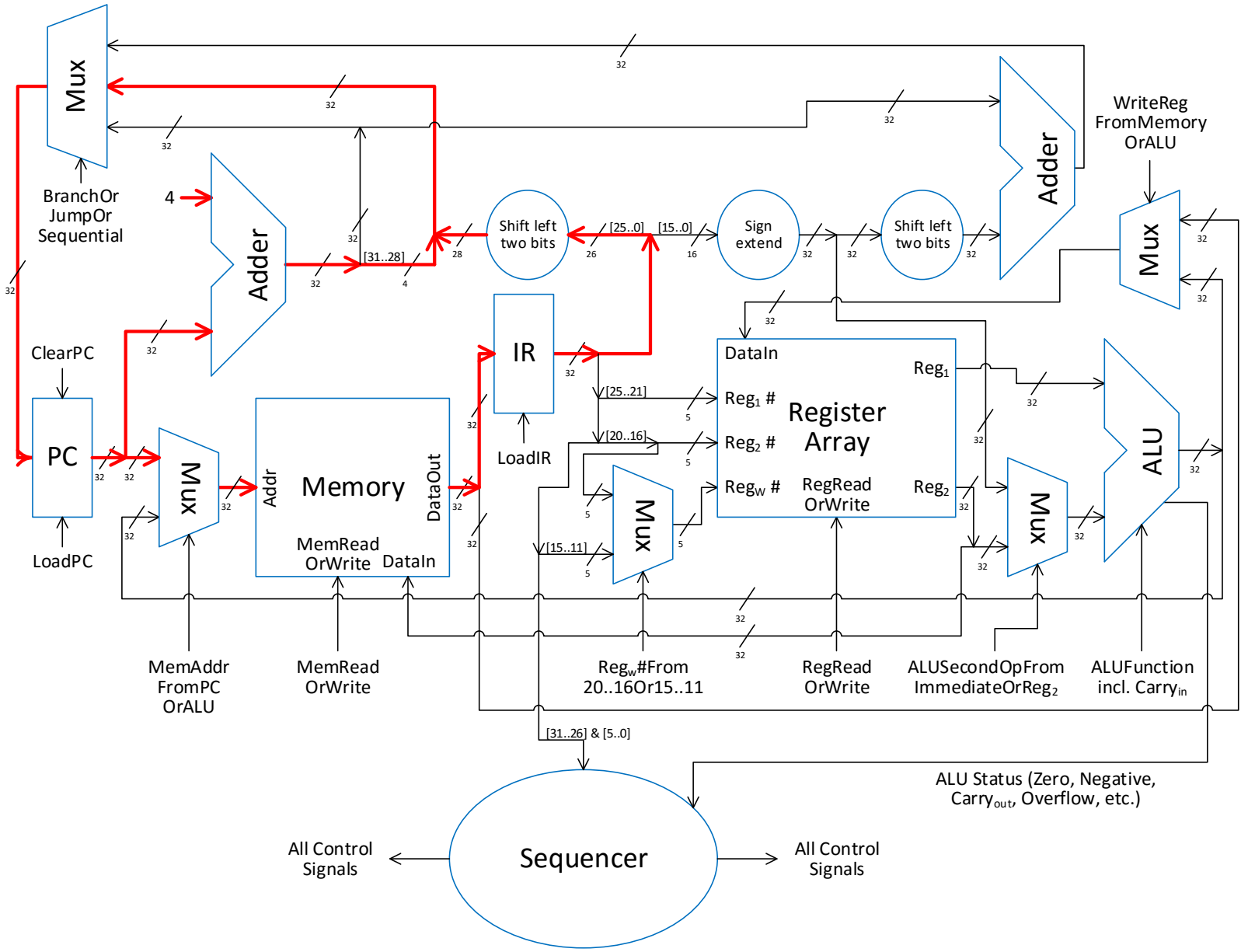
Branch I-Type Instructions


Mux
BranchOr
JumpOr
Sequential
4
Adder
[31..28]
Shift left
two bits
[25..0]
[15..0]
Sign
extend
Shift left
two bits
Adder
WriteReg
FromMemory
OrALU
Mux
ClearPC
PC
LoadPC
Mux
Addr
Memory
DataOut
MemRead
OrWrite
DataIn
IR
LoadIR
[25..21]
[20..16]
[15..11]
Mux
DataIn
Reg1 #
Reg2 #
Regw #
Register
Array
RegRead
OrWrite
Reg1
Reg2
Mux
ALU
MemAddr
FromPC
OrALU
MemRead
OrWrite
Regw#From
20..16Or15..11
RegRead
OrWrite
ALUSecondOpFrom
ImmediateOrReg2
ALUFunction
incl. Carryin
[31..26] & [5..0]
ALU Status (Zero, Negative,
Carryout, Overflow, etc.)
All Control
Signals
Sequencer
All Control
Signals

Branch and Link I-Type Instructions

- Branch and Link I-Type instructions are not supported by our block diagram
- Missing components
  - No way to force use of GPR 31 ($ra) as the Reg_w # of the Register Array
  - No path for the return address (incremented PC) to DataIn of the Register Array

Jump J-Type Instruction


Mux
BranchOr
JumpOr
Sequential
4
Adder
[31..28]
Shift left
two bits
[25..0]
[15..0]
Sign
extend
Shift left
two bits
Adder
WriteReg
FromMemory
OrALU
Mux
ClearPC
PC
LoadPC
Mux
Addr
Memory
DataOut
MemRead
OrWrite
DataIn
IR
LoadIR
[25..21]
[20..16]
[15..11]
Mux
DataIn
Reg1 #
Reg2 #
Regw #
Register
Array
RegRead
OrWrite
Reg1
Reg2
Mux
ALU
MemAddr
FromPC
OrALU
MemRead
OrWrite
Regw#From
20..16Or15..11
RegRead
OrWrite
ALUSecondOpFrom
ImmediateOrReg2
ALUFunction
incl. Carryin
[31..26] & [5..0]
ALU Status (Zero, Negative,
Carryout, Overflow, etc.)
All Control
Signals
Sequencer
All Control
Signals

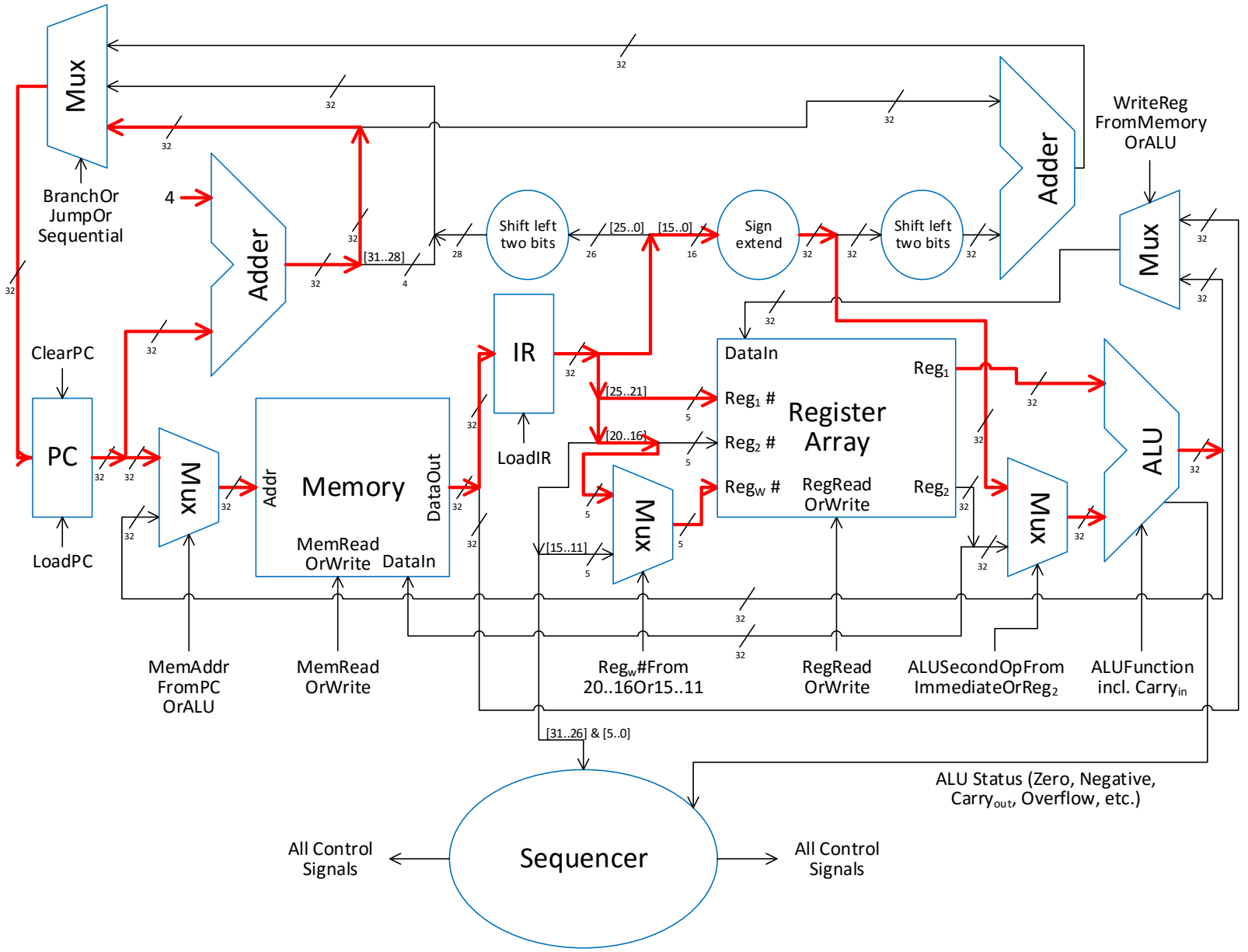
Jump and Link J-Type Instructions

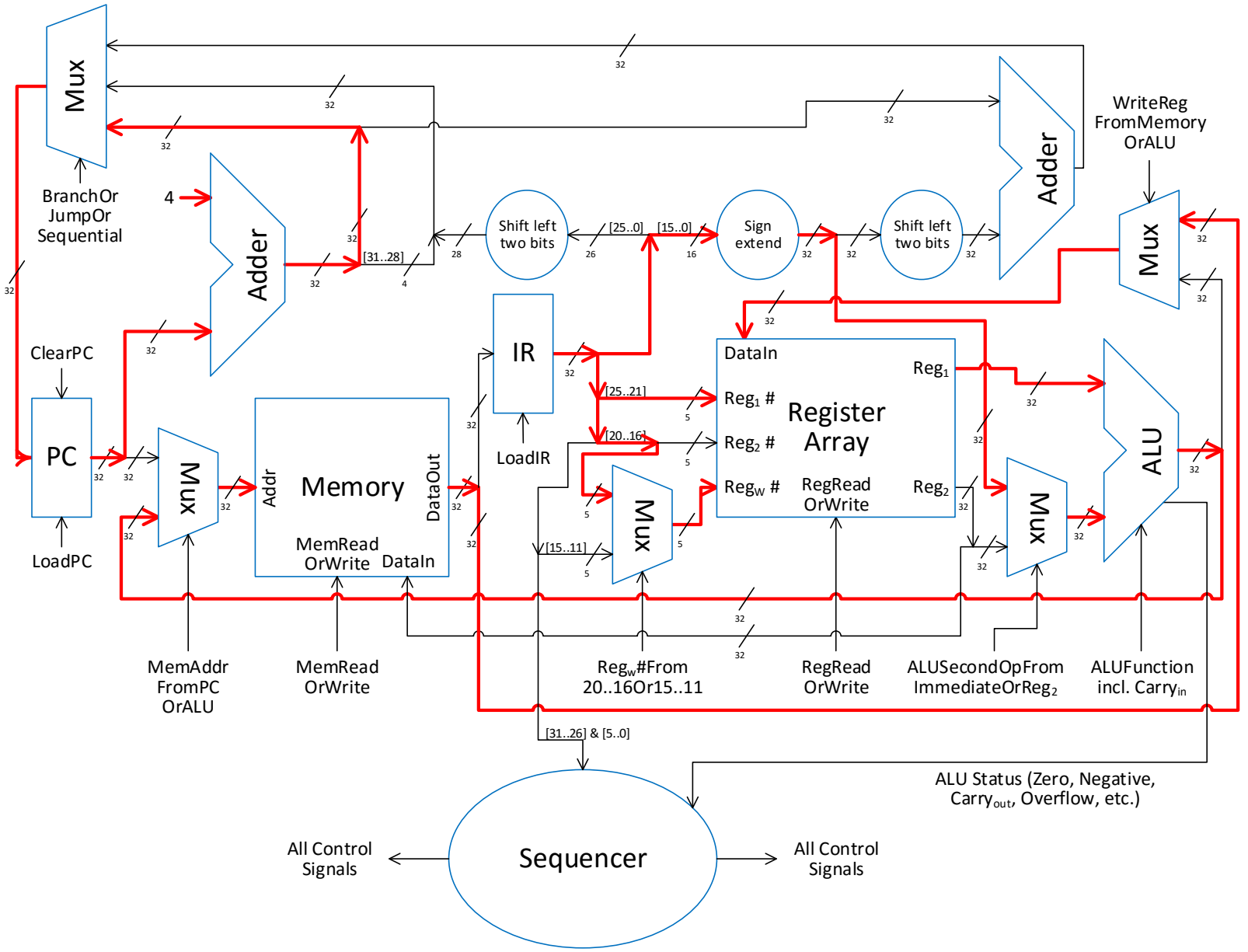
- Jump and Link J-Type instructions are not supported by our block diagram
- Missing components
  - No way to force use of GPR 31 ($ra) as the Reg_w # of the Register Array for the JAL instruction
  - No path for the return address (incremented PC) to DataIn of the Register Array

Jump Register Instructions

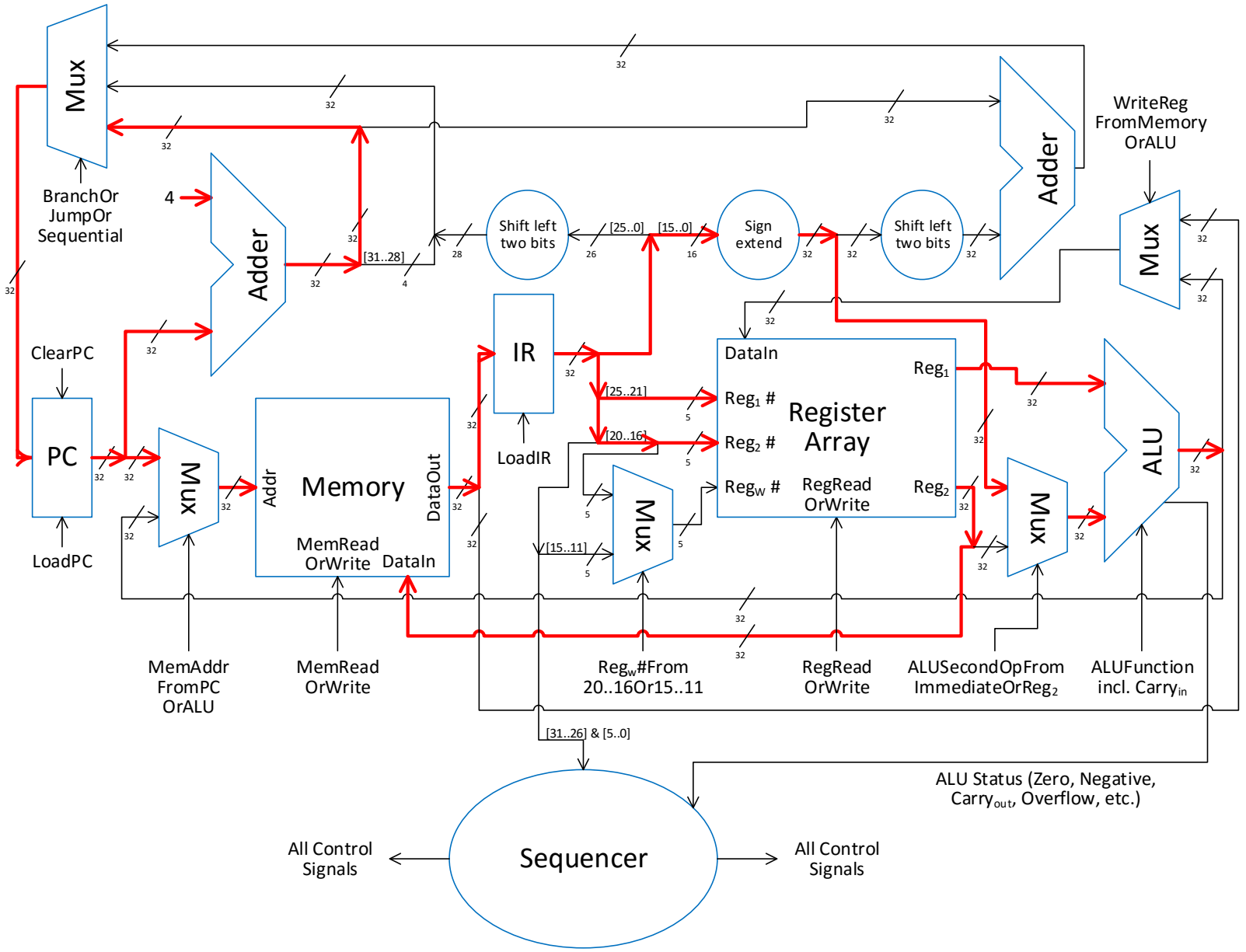
- Jump Register instructions are not supported by our block diagram
- Missing component for JR and JALR
  - No path from Reg_1 to the input to the PC
- Missing component for JALR
  - No path for the return address (incremented PC) to DataIn of the Register Array

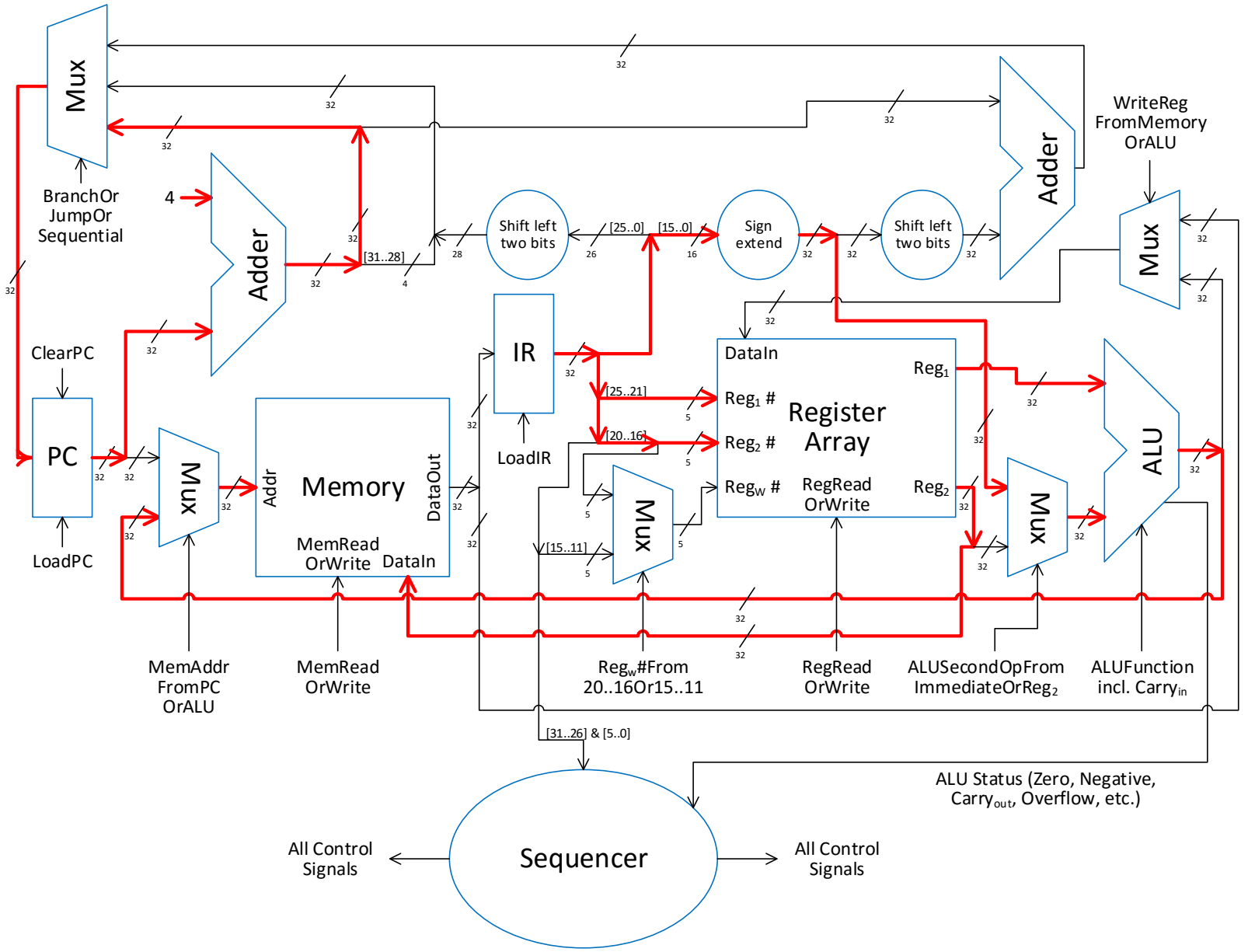
Load I-Type Instructions


Mux
BranchOr
JumpOr
Sequential
4
Adder
[31..28]
Shift left
two bits
[25..0]
[15..0]
Sign
extend
Shift left
two bits
Adder
WriteReg
FromMemory
OrALU
Mux
ClearPC
PC
LoadPC
Mux
Addr
Memory
DataOut
MemRead
OrWrite
DataIn
IR
LoadIR
[25..21]
[20..16]
[15..11]
Mux
DataIn
Reg1 #
Reg2 #
Regw #
Register
Array
RegRead
OrWrite
Reg1
Reg2
Mux
ALU
MemAddr
FromPC
OrALU
MemRead
OrWrite
Regw#From
20..16Or15..11
RegRead
OrWrite
ALUSecondOpFrom
ImmediateOrReg2
ALUFunction
incl. Carryin
[31..26] & [5..0]
ALU Status (Zero, Negative,
Carryout, Overflow, etc.)
All Control
Signals
Sequencer
All Control
Signals


Mux
BranchOr
JumpOr
Sequential
4
Adder
[31..28]
Shift left
two bits
[25..0]
[15..0]
Sign
extend
Shift left
two bits
Adder
WriteReg
FromMemory
OrALU
Mux
ClearPC
PC
LoadPC
Mux
Addr
Memory
DataOut
MemRead
OrWrite
DataIn
IR
LoadIR
[25..21]
[20..16]
[15..11]
Mux
DataIn
Reg1 #
Reg2 #
Regw #
Register
Array
RegRead
OrWrite
Reg1
Reg2
Mux
ALU
MemAddr
FromPC
OrALU
MemRead
OrWrite
Regw#From
20..16Or15..11
RegRead
OrWrite
ALUSecondOpFrom
ImmediateOrReg2
ALUFunction
incl. Carryin
[31..26] & [5..0]
ALU Status (Zero, Negative,
Carryout, Overflow, etc.)
All Control
Signals
Sequencer
All Control
Signals

Store I-Type Instructions


Mux
BranchOr
JumpOr
Sequential
4
Adder
[31..28]
Shift left
two bits
[25..0]
[15..0]
Sign
extend
Shift left
two bits
Adder
WriteReg
FromMemory
OrALU
Mux
ClearPC
PC
LoadPC
Mux
Addr
Memory
DataOut
MemRead
OrWrite
DataIn
IR
LoadIR
[25..21]
[20..16]
[15..11]
Mux
DataIn
Reg1 #
Reg2 #
Regw #
Register
Array
RegRead
OrWrite
Reg1
Reg2
Mux
ALU
MemAddr
FromPC
OrALU
MemRead
OrWrite
Regw#From
20..16Or15..11
RegRead
OrWrite
ALUSecondOpFrom
ImmediateOrReg2
ALUFunction
incl. Carryin
[31..26] & [5..0]
ALU Status (Zero, Negative,
Carryout, Overflow, etc.)
All Control
Signals
Sequencer
All Control
Signals


Mux
BranchOr
JumpOr
Sequential
4
Adder
[31..28]
Shift left
two bits
[25..0]
[15..0]
Sign
extend
Shift left
two bits
Adder
WriteReg
FromMemory
OrALU
Mux
ClearPC
PC
LoadPC
Mux
Addr
Memory
DataOut
MemRead
OrWrite
DataIn
IR
LoadIR
[25..21]
[20..16]
[15..11]
Mux
DataIn
Reg1 #
Reg2 #
Regw #
Register
Array
RegRead
OrWrite
Reg1
Reg2
Mux
ALU
MemAddr
FromPC
OrALU
MemRead
OrWrite
Regw#From
20..16Or15..11
RegRead
OrWrite
ALUSecondOpFrom
ImmediateOrReg2
ALUFunction
incl. Carryin
[31..26] & [5..0]
ALU Status (Zero, Negative,
Carryout, Overflow, etc.)
All Control
Signals
Sequencer
All Control
Signals