# Tutorial 11

## A performance oriented $\beta$-machine

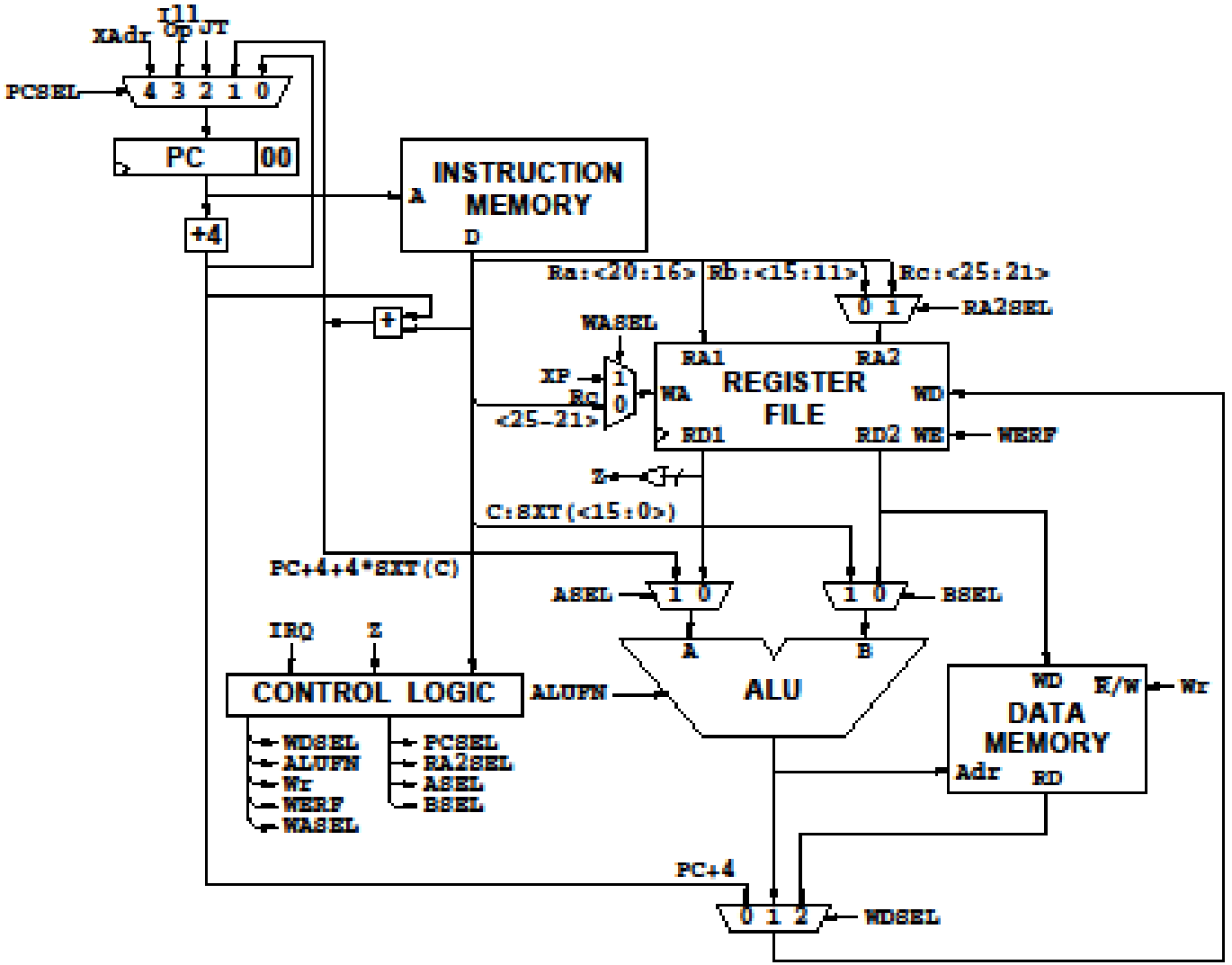
December 13, 2016

# ULg0X is slow !

- Instruction are implemented using microcode in ULg0X machines
- Microcode is a step-by-step implementation of the instruction
- Each used phase of the microcode consumes one clock cycle
- At worst:
  - 16 clock cycles on ULg01 and ULg02 per instruction
  - 32 clock cycles on ULg03 per instruction (when there are only cache hits)
  - Even worse on cache misses as we have to execute a whole procedure !
- Can we do better ?

# We can do better !

- Solution: the **performance oriented $\beta$-machine** !
- Everything is executed in one clock cycle
- No more microcode

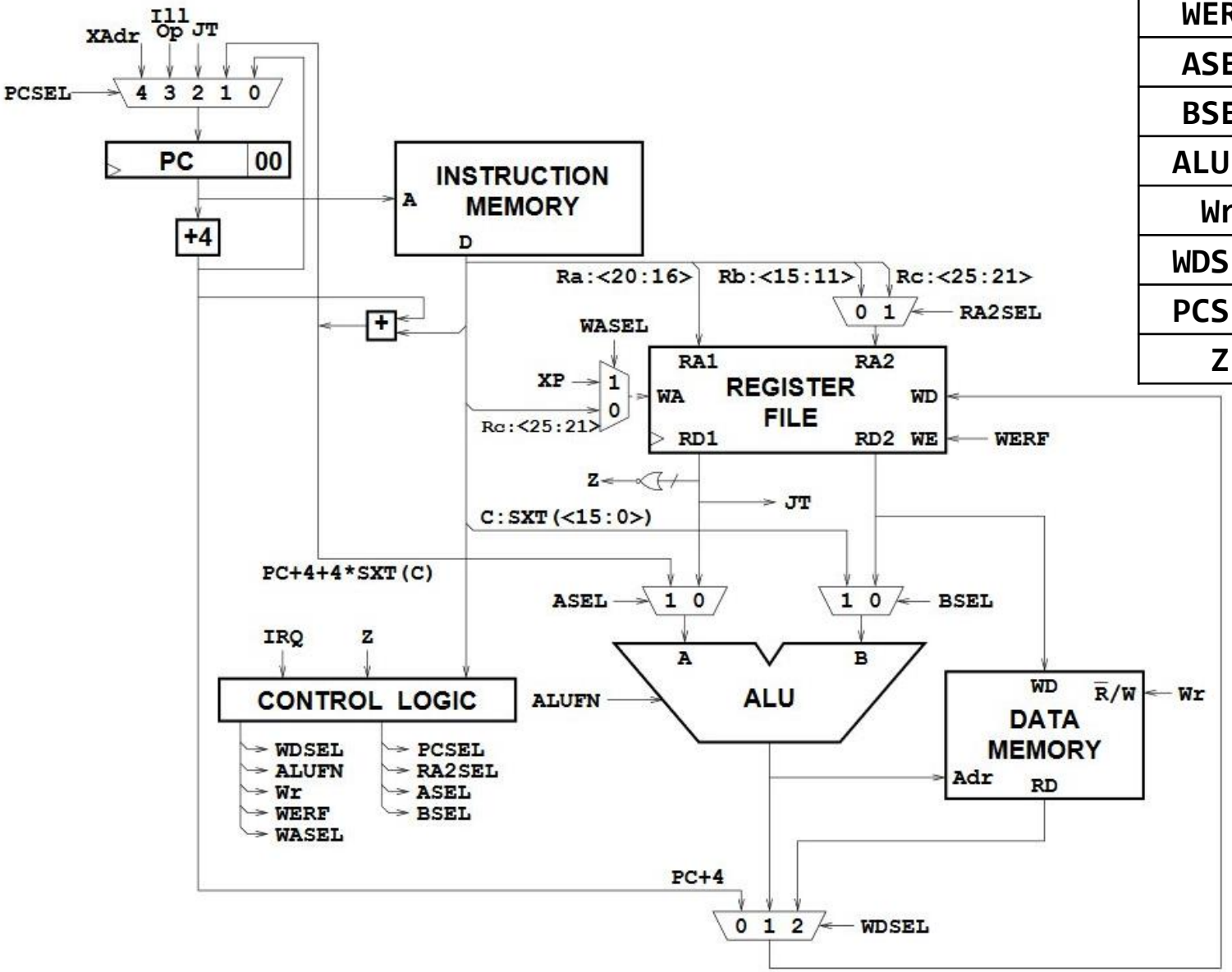


# Performance oriented $\beta$-machine

- Several data transfer path
- Components' inputs are connected to multiplexers (can use signals from several sources)
- **Register file**: two reads and one write at the same time on the registers!
- Managed by a control logic component (implemented by a ROM or a PLA)
- The control logic controls the inputs and outputs of the components
- Data memory and instruction memory viewed as seperate (implement by caches)

# Limitations

- No supervisor mode
- Virtual memory not taken into account
- The circuit has a **long stabilization time**. The machine must operate at a smaller frequency than ULg0X.
- A solution to this last limitation is a pipeling (more about this next week)




| Signals | Meaning |
|---|---|
| **RA2SEL** | Second register read address selection |
| **WASEL** | Register write address selection |
| **WERF** | Register write enabled |
| **ASEL** | ALU A input selection |
| **BSEL** | ALU B input selection |
| **ALUFN** | ALU function selection |
| **Wr** | Memory write enabled |
| **WDSEL** | Register write input selection |
| **PCSEL** | PC update input selection |
| **Z** | 1 if RD1 is zero, 0 otherwise |