 [Translated from Greek to English - www.onlinedoctranslator.com](https://www.onlinedoctranslator.com/en/?utm_source=onlinedoctranslator&utm_medium=docx&utm_campaign=attribution)

12/5/2018

# 5th Laboratory Exercise

**CONVERTING CHARIS-4 INTO A PIPELINE PROCESSOR**

Group LAB31235515

|  |
| --- |
| *CHRISTOS ZISKAS 2014030191* |
| *ANTONIS ANTONAKAKIS 2014030160* |

## Purpose of laboratory exercise

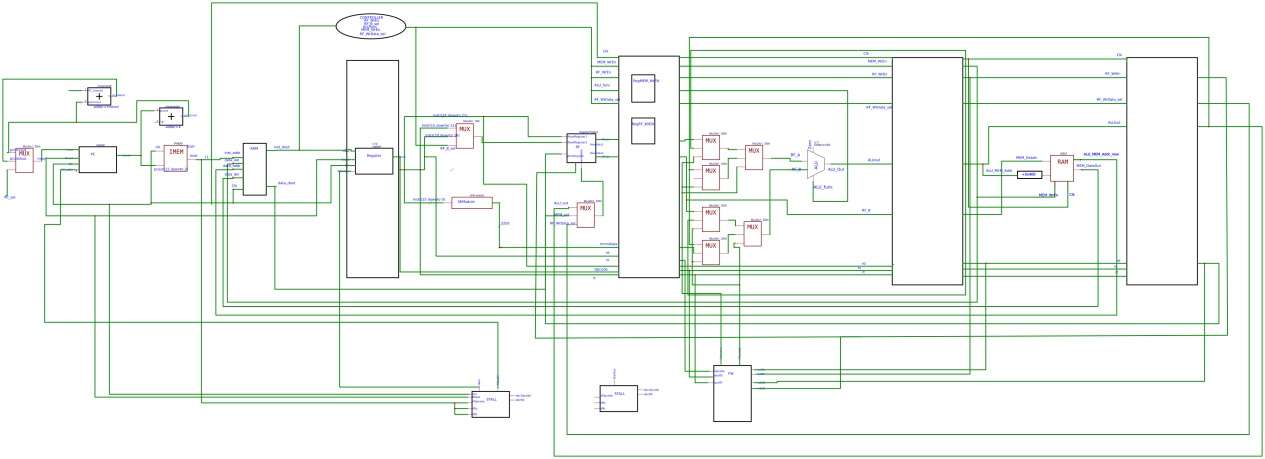
The reconfiguration of multi-cycle processor functionality

in a pipeline processor which upgrades the processor by giving it further time performance characteristics. It falls short in terms of the command architecture it executes but there is a gain in time. The design is redefined with the influx of new elements of delay and prediction of the contents to be executed, as well as the introduction of combinational logic and non.

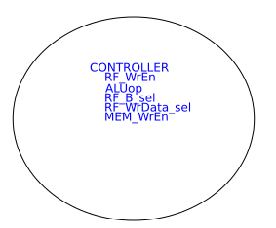
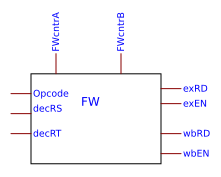
## Preparation

To complete the design of the pipeline processor, it is necessary to correct the control in order to settle the remaining commands as the control signals that direct the datapath are recognized in a cycle and simultaneously run through the various pipes as the cycles pass. Each command remains in the control for one cycle.. In the new control, the logic of most commands is removed and it concerns the commands li,add,lw,sw. the clock is missing. The rf\_b\_sel is exported directly to the rf to channel the necessary results to the outputs of the rf for each command. There are 4 types of pipes: IF/DEC, DEC/EX, EX/MEM, MEM/WB. IF/DEC is a register to store the fetched instruction. The DEC/EX pipe is a set of registers that keeps the results from reading the rf outputs, the write position, the rt,rs addresses, the immidiate, opcode as well as the signals that the control outputs for the command. The EX/MEM pipe uses fewer registers as the alu function(alu func) and Immidiate have been specified as inputs to alu to continue the write. So registers for the rest of the control signals (RF\_En, Mem\_En, RF\_Datasel), the result from alu, the result from reading the second output of rf, the recording position, the addresses rt,rs. In the MEM/WB pipe, enable and the memory address are removed as they have been assigned to ram. Finally the command is completed after five cycles while at the same time new commands are executed which are stored in previous stages and the same procedure is followed.. The design of the command path as fsm is recommended below:

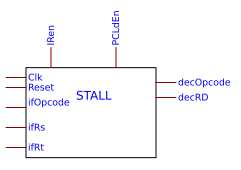
StepDatapath



Control Unit (Control) Forward Unit

Stall unit



## Description

Datapath and control functionality is different in the new version. There are conversions due to pipeline elements to execute and store instructions from the processor in parallel.

The decoding stage (decstage) is maintained with the only difference that the recording position appears as input so that when the correct storage position of the instruction is set in the MEM/WB pipe, the corresponding position is also applied to the register file. This input is channeled by the WB -rd. The end of the registration is configured after the time interval of 5 cycles. The extension item is reduced with less functionality due to the removal of existing commands.

The calculation stage (alustage) accepts conversions that help to execute the commands either in correct parallelism or through advance and delay. The signals aluabinsel, alubbinsel and their multiplexers are removed. The internal control unit alu control (handled by aluop and opcode) is removed as its function becomes redundant. The operation that alu employs is only about addition for existing commands.

The pipes consist of a set of registers so that at any stage of the path the command to be registered is attended and new commands are returned to the fetch again. The dec/ex stage is additionally fed by the control signals so that the contents developed by the command direct the various elements of the datapath so that the correct signals are routed to address and write. At each stage, the signals stored from pipe to pipe decrease.

(alu func to alu, rfbsel goes directly to rf without going into the pipe, memwr to memory etc)

The datapath, in turn, has received variations. 2 4-to-1 multiplexers are introduced to deal with the forward effect (some instructions use one register as a register while the next ones use the same as the destination or source register). So 3 positions are covered by the result of alu in the pipe dec/ex and ex/wb as well as the output of rf , one for each multiplexer (rfa,rfb).

The first multiplexer accepts zeros as the fourth port while the second multiplexer the immidiate. The output signals of the forward unit, one for each multiplexer, are recommended as a control signal for the multiplexers.

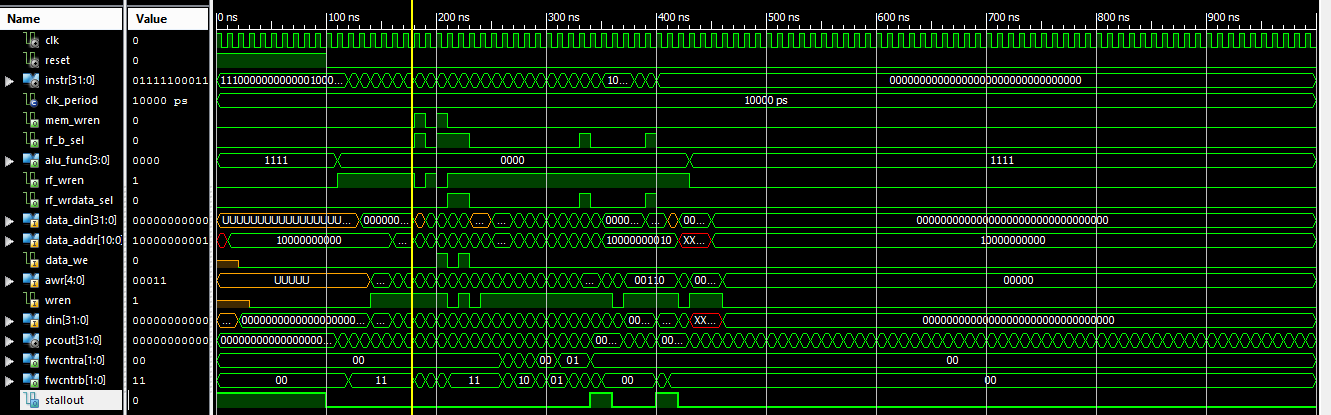
The memory does not undergo new modifications.

Forward unit and stall elements are included in the design. Forwarded inputs from alu to multiplexers are selected when an instruction uses as a register write that is later used as a source by an instruction in the datapath at a previous stage. The forward delivers the appropriate signals to the multiplexers while accepting as inputs the source registers from the dec/ex stage checking them with the register position in the pipe ex/mem ,mem/wb. For the input of immidiate to alu, it is sufficient that the command add has not been given. In any other case the immidiate is given. The stall occurs when an add command is given after a lw command. The treatment of this problem is corrected by delaying the command (the enable signal of the pc as well as the if/dec Pipe is cut off). Thus, the Lw command manages to cross the datapath and due to the delay create gaps (bubbles) in the intermediate stages. Runs as fsm with initial state, lden open and Pipe enable . By checking that the add command occupies a register of the lw command, the control signal of the pc counter and the pipe is disabled and maintained for the required delays. For delay 1 the forward is done by ex/mem pipe while for delay 2 it is done by mem/wb.

Control is simplified with fewer states to make signals respond to system commands while the clock is absent. So the requested signals are exported directly by entering the command in the control and after one cycle they are stored in the corresponding pipe and follow their correct path.

## Waveforms-Simulation

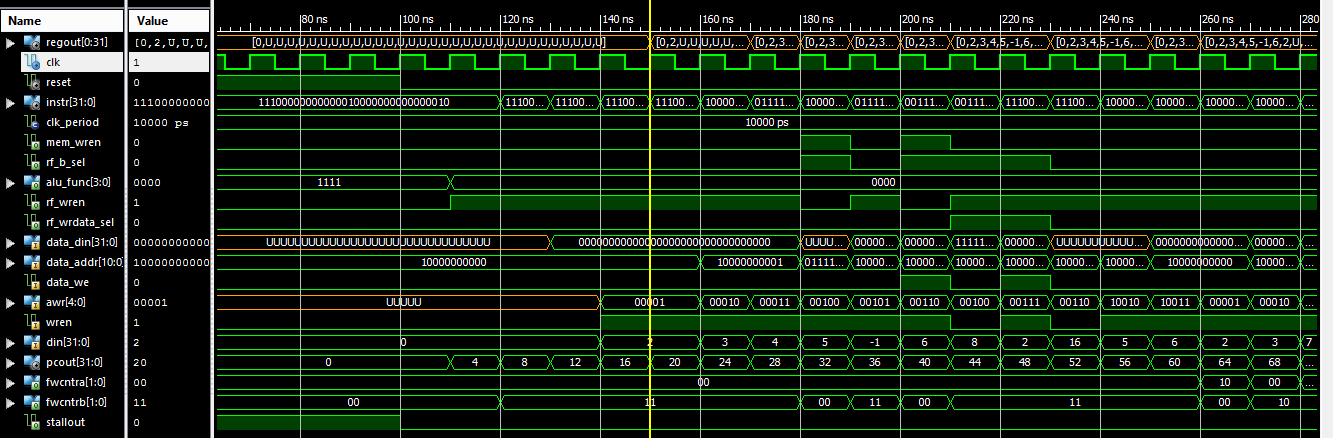
Pipeline processor waveforms are shown.



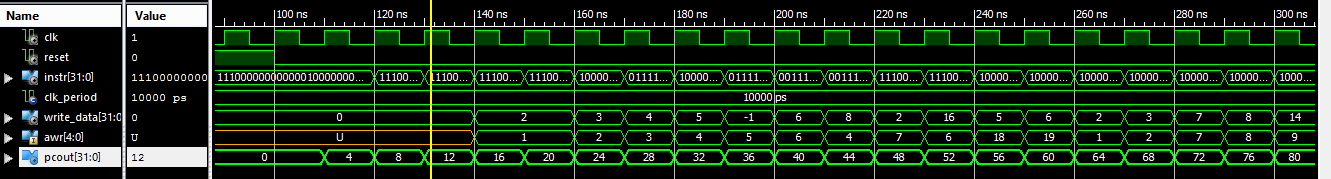


Each command takes 5 cycles except for the store command

All processor elements enter the datapath so that there is an immediate response of the memory elements after one cycle (otherwise they would need to

recalculate new delays for storage).

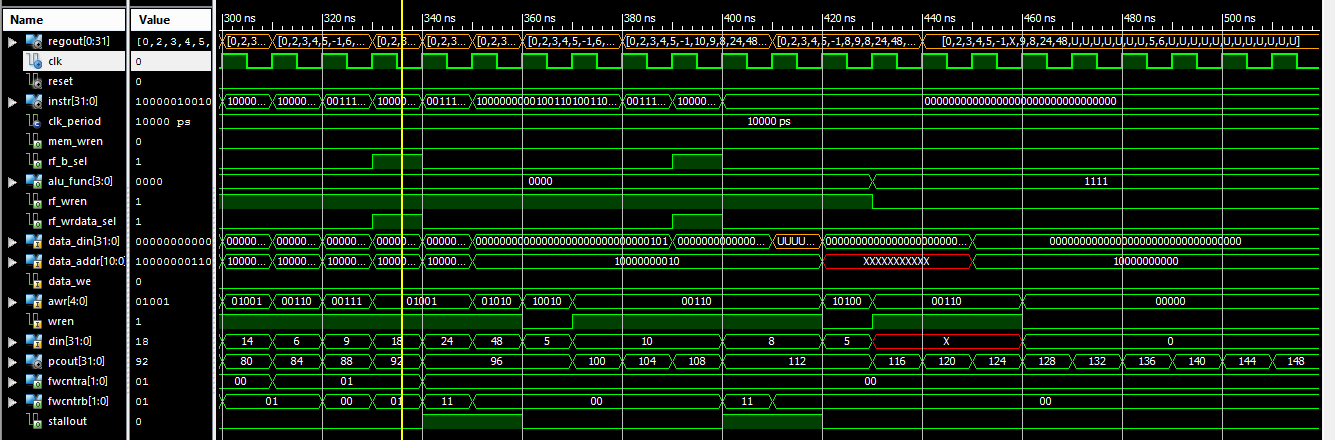
Observing the waveform, it is recognized that by removing the reset from the circuit, the rf recording is applied after 5 cycles (||) while the rf write signal appears after 4 cycles. Each instruction is executed within one cycle as the control outputs the control signals, stores them in the pipe, and the next instruction arrives. The control essentially releases the signals for each command while at the same time as they move along the path they end up in the objects that handle them in order to reach their correct results. If a signal or the result is not processed, the direction to some element is then led directly to the next pipe. Finally, the result ends up in the register to be written or in the memory for the store command. In addition, one cycle is required to adapt the instruction to the pipe if/dec and control .Then the enable for the li instruction is activated one cycle later. Commands are fetched from the circuit and routed along with the control signals. Checking the store instruction verifies its execution within 4 clock cycles.



The outputs rendered are the desired ones

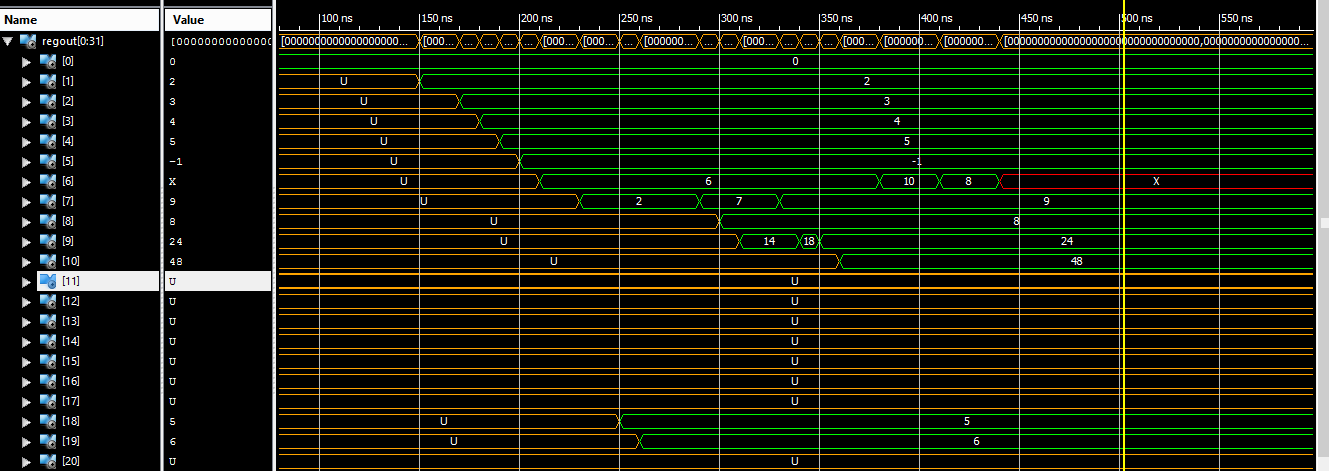


As the first results come in, the forward application is implemented by the registers (the signals of the multiplexes change). The results are the desired ones. Next is the stall for which there is a kind of error as due to a timing problem the correct data and control signals are not routed at the correct time so that the circuit does not have time to absorb the entry of register 20 and be indifferent (XXXXXX…) the contents of the register 7 (various attempts were made to find a solution to this problem: introducing an extra register at the fetch point for further delay, converting the stall from sequential to combinational logic with signal generation from stall to lden inverted as in the if/dec pipe, inserting a multiplexer at the point of dec/ex so that when the system is in a stall the wren changes from 1 to 0 to cancel the "garbage" commands due to the " "bad" timing.

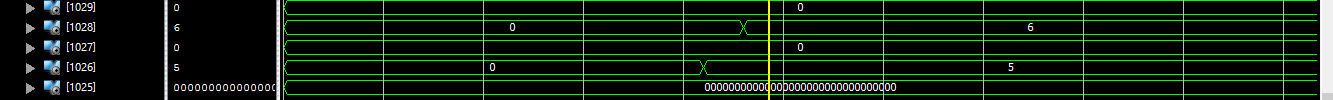


Verifying the results from the pc counter, the commands are adequately and successfully routed to the circuit (there is an error). The results are saved

to the registrars



as well as in memory



## Conclusions/Problems

The lab exercise is a natural consequence of understanding the differences between a multi-cycle processor and a single processor. The parallel execution of the commands contributes to the faster management of the system accompanied by timing problems. The design of such a circuit overlaps issues of optimal performance and speed as opposed to multi-cycle processors that deal with easier problems over a longer period of time.