**Computer Architecture 2023**

**Exercise Session Report**

Group: 20

Name and student number:

Jitse Kennes r0798232

Giel Ooghe r0790882

Date: 20/04/2023

**Part 1: Fill in the Design Metrics**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Implementation*** | ***Simulation*** | ***Backend*** | | | | | | |
| *Synthesis* | | | | *Floorplan* | *Signoff* | |
| *# of Cycles for MULT\** | *Design*  *area (μm2)* | *Critical path*  *(ns)* | *Maximum operating frequency (MHz)* | *Minimal time to execute the MULT (ns)* | *Die area (μm2)* | *Area utilization (%)* | *Maximum operating frequency (MHz)* |
| *1. Single Cycle\*\** | *1* | *508312* | *9.85* | *101.5* | *9.85* | *\*\*\** |  |  |
| *2. Single Cycle with Multiplication* | *1* | *617354* | *12.97* | *77.1* | *12.97* |  |  |  |
| *3a. Basic Pipelined* | *40* |  |  |  |  |  |  |  |
| *3b. Basic Pipelined (50MHz, Area 2)* | *40* | *635498* | *18.10* | *55.25* | *724* |  |  |  |
| *3c. Basic Pipelined (50MHz, preset 2)* | *40* |  |  |  |  |  |  |  |
| *4. Pipelined with hazard logic* | *25* | *639065* | *0.88* |  |  |  |  |  |
| *5. Advanced acceleration* | *784* | *651415* | *56.52* | *17.7* | *44311* |  |  |  |

*\* The program*

* *MULT1 is for “1. Single cycle”;*
* *MULT2 is for “2. Single Cycle with Multiplication” and “3a/3b. Basic Pipelined”;*
* *MULT3 is for “4. Pipelined with hazard logic”;*
* *MULT4 (or your modified version of it) is used for “5. Advanced acceleration”.*

*\*\* Implementations 1, 2, 3a, 4 and 5 have a clock frequency of 10MHz and is synthesized under the default strategy “AREA 0”.*

*\*\*\* The grey regions are* ***not required*** *to fill in.*

*Presets 1 & 2 must be replaced by your synthesis strategy after performing the Synthesis Exploration step (e.g., DELAY 0-4/AREA 0-3).*

**Part 2: Answer the Questions based on Your Designs**

***Questions:***

1. *What is the critical path for the single-cycle processor (simple program)? Which operation does this critical path stand for?*

*This is probably a load instruction.*

1. *For the single-cycle processor, what are the top-5 components in the area cost? After adding multiplication support, what are the top-5 components in the area cost then? What has the changed? What is your explanation for this?*

1. *What is the critical path after you support the MULT operation in the single-cycle processor? Is there anything changed? What is your explanation for this?*
2. *Regarding to your findings in Q2 & Q3, do you think adding MULT operator support a good choice for every microprocessor? Please elaborate your answer.*

*If your area availability is restricted and speed is less of a concern then it would not be a great idea since a dedicated MULT operator trades off area for speed.*

1. *What is the critical path in the pipelined design? How does the critical path length change comparing to the single-cycle designs? How about the area? Try to analyze the benefits of processor pipelining according to your findings.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. *What techniques did you apply to accelerate MULT4? Explain under what conditions/type of workload you techniques will have the maximum/minimum performance gain.*

*We reordered the instructions in the imem file. This avoids a stall between a load and a MUL operation. If there are a lot of data hazards after loads because of instantaneous use of loaded variables. This would create 1 stall even with forwarding and thus we avoid this by reordering the instructions.*

1. *(Openlane Flow) Please explain the following terms of the backend flow: PDK, LEF, CTS, STA, GDSII. Which PDK are we using?*

*PDK: Process design kit*

*LEF: Library Exchange Format*

*CTS: clock tree synthesis*

*STA: static timing analysis*

*GDSII: Graphic Design System 2*

*We are using sky130A as PDK*

1. *(Openlane Flow) We used lots of open-source tools for the backend. What is the difference and relationship between OpenRoad, OpenRAM and OpenLane? How are they used in our backend flow?*

*openRoad -> open source layout generation (RTL to GDS)*

*openMem -> open source memory compiler*

*openLane -> scripting software for use of openRoad, openRam …*

*openLane runs all the other tools*

1. *(Openlane Flow) The major steps in the backend flow this exercise session focuses on are Synthesis, Floorplan, Placing and Routing. Have a look at the “runs/results/” folder. What are the output files of each step? What are the similarity and differences?  
   Synthesis: Verilog files and sdf file*

*Floorplan: def file and odb file*

*Placement: Verilog files, odb file and def file*

*Routing: same as placement but with mca folder with lib and sdf files*

*Sdf are spatial data files, def design specific files, odb from OpenOffice Base and Verilog contain RTL info.*

*Verilog files will thus contain data about the logical connections while the sdf files more over the physical placement. The others could be more metadata.*

1. *(Openlane Flow) Try to compare between the three full-flow scenarios. What is the impact of the target clock frequency? What is the impact of the synthesis strategy? Please discuss the tradeoff in terms of speed and area, i.e. under what scenario will you choose which design.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**Part 3: Post your design photos**

*Following the step “****Analysis and Discussion #5****” in Jupyter Notebook, take a screenshot of the final die photo.*

*Photo 1: Signoff - Basic Pipelined (10MHz)*

*A close-up of a computer chip

Description automatically generated with medium confidence*

*Photo 2: Signoff - Basic Pipelined (50MHz)*