**Report Computer Architecture 2022**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Implementation* | *Area*  *(with SRAM / without SRAM)* | *Critical Path (ns)* | *Maximum Operating Frequency* | *Number of Cycles for program MULT\** | *Minimal time to execute the program MULT* |
| *Single Cycle* | *408393.738746* | *69.03* | *14.43 MHz* | *2300* | *158 769 ns* |
| *Single Cycle with Multiplication Support* | *419684.594178* | *63.92* | *15.64 MHz* | *40* | *2556.8 ns* |
| *Pipelined* | *423803.251915* | *45.53* | *21.96 MHz* | *40* | *1821.2 ns* |
| *Pipelined with hazard and stall logic* | *424449.540120* | *51.64* | *19.36 MHz* | *25* | *1291 ns* |
| *Advanced acceleration* | *424446.490376* | *42.98* | *23.26 MHz* | *890* | *38 252.2 ns* |

*\* The program MULT1 is used for “Single cycle”, MULT2 for “Single Cycle with Multiplication Support” and “Pipelined”, MULT3 is used for “Pipelined with hazard and stall logic”, MULT4, or your modified version of it, is used for “Advanced acceleration”.*

*Questions:*

* *For the single cycle processor, which kind of instruction would stimulate the critical path found? How would you improve it without adding any pipe stage?*

*The load instruction results in the longest critical path and thus sets the clock speed. A way to improve this.*

* *For the single cycle processor, which resources constitute most part of the gates? What is your explanation for this distribution? Is it possible to reduce the number of gate cells?*

*The data memory constitutes for 305 707.4986 of the area. This is because memory is always a large contribution, as you need multiple gates per bit, multiple bits per word, and multiple words.*

* *Is the critical path affected when hardware support for multiplication is added to the single cycle processor? What is your explanation for this? Do you know any multiplier implementation that can improve timing?*

*In our case the critical path is reduced. A possible explanation is that the synthesis tool can organize the architecture in a better way, thus reducing the critical path. We don’t know what the internal implementation of the multiplier in our case. One possible implementation is a Wallace tree multiplier where it sums partial products in stages until two numbers are left.*

* *Is adding hardware support for multiplication a good choice for every microprocessor? Motivate your answer.*

*No, because it increases the critical path. If multiplication is not required, or rarely used, it is better to not implement it to benefit more from the reduced critical path and thus a higher possible clock frequency. Including the multiplication also increases the size, and thus the cost of the processor.*

* *How much larger is the pipelined implementation compared to the single cycle processor? What is the main cause for its increase? How is the critical path affected when we pass from a single cycle processor to a pipelined implementation?*

*The pipelined implementation is about 4% larger. The main cause for this increase is the pipeline registers added between each stage. The critical path decreases.*

* *Considering the critical path found for the pipelined processor, how would it be possible to increase the performance of the system? Would your solution significantly speed up the core? Also, what will be the new critical path?*

*The critical path is in the EXE stage and goes through the ALU. So a way to increase the performance is to improve the ALU implementation and thus reduce the critical path. This will not give a significant benefit. Another way to speed up is to execute two instructions in parallel as seen in the course. This will result in a speed up, but not with the expected factor 2.*

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

*We applied the data forwarding and load hazard detection of the previous session. We further added the baseline implementation with branch not-taken prediction and the flushing of instructions. This will have a maximum performance if most branches are not taken, for example with for-loops that are repeated many times.*

* *Is the addition of hardware improvements, like pipelining, correlated with higher power consumption? How can we assess if a specific modification to our processor improves or diminishes the energy efficiency of the system?*

*In this case it will consume more power than the single cycle processor, as each stage of the pipeline is active in the general case. This means that at least the same power as the single cycle processor is used, in addition to the possible power used by the registers and forwarding/hazard detection units.*