AN ENERGY EFFICIENT SINGLE CYCLE PROCESSOR

Santosh Krishnan (PID A53071285) & Devashish Deshpande (UID UO6859098)

THE DESIGN CHALLENGE

In today's digital age there is a huge demand for small portable devices and gadgets like smartphones and smartwatches that have large computational ability. However the main challenge of designing these devices is that the processors that power these devices must consume very low energy as these devices are powered by batteries. As part of our ECE 260A course we designed the functional part of a basic single cycle processor that generates a six bit result and optimize the architecture to minimize energy average energy consumed per operation while maintaining a throughput of 10ns.The processor included program counter, a memory to store instructions, an instruction decoder and an ALU that performs addition, subtraction, multiplication, logical OR, logical AND, setless which identifies which of the inputs are larger. Various power saving techniques were used which will be elaborated below.

BASIC ARCHITECTURE

In our processor, we have implemented a two stage pipelined processor. In the first stage the program counter gives the memory address of instruction to be executed. The memory accepts the address and gives a eleven bit instruction where the first three bits(MSB's) of the instruction are the opcode, the next four bits is the value of variable X and the last four bits are of variable Y. In the second stage, the instruction decoder accepts the opcode and generates control signals for the multiplier, logical OR, logical AND, and the adder-cum- subtractor blocks. The outputs of all the blocks are given to 6 multiplexers which give the required result.

We decided to first design each block with reasonable specs and then put all the blocks together as soon as possible. Once the processor was functional, we decided to optimize the blocks for overall better performance. The first challenge was to ensure that the processor executes the instruction correctly with a throughput of 10ns, however the main challenge of this project was to reduce the power consumption and still be able to meet the throughput requirement.

INNOVATIVE ASPECTS ABOUT OUR DESIGN

One of the innovative design we have in our processor is the ROM. We decided to employ the NAND ROM architecture. While working on it, we realized that PMOS size determines current through the circuit and in order to minimize the current consumption, we need to minimize the W/L ratio of the PMOS transistors. Secondly, the NAND ROM is a ratioed circuit and it naturally needs a very high pull down strength. To achieve this, we had to bump up the NMOS W/L ratios. There was a tradeoff: Higher the NMOS ratios were, better was the pull down strength but more was power consumption. Memory is one of the primary reasons for excess power consumption as there is a short circuit current through it most of the time. In order to reduce the short-circuit current, we decided to clock the memory circuit. In other words we used a dynamic circuit instead of a pseudo NMOS in the NAND ROM memory. By doing so we managed to reduce energy by almost 40% for VDD equal to 1V. However we weren't able to scale VDD to very low values thereby limiting the amount of power we could save. The reason for that was that our critical path was very long and half of our clock period was wasted by our memory. To get around this problem we pipelined our design into two stages in order to shorten our critical path. We also tried for power gating but we found out that the savings in power was negligible. Another technique we used to save energy was to give low state inputs to the unused sub blocks in the ALU. The advantage of using such a technique was that out output multiplexer became very simple to design as the outputs from the unused blocks were always zero. Thus we could use a simple 5 input OR gate as the multiplexer.

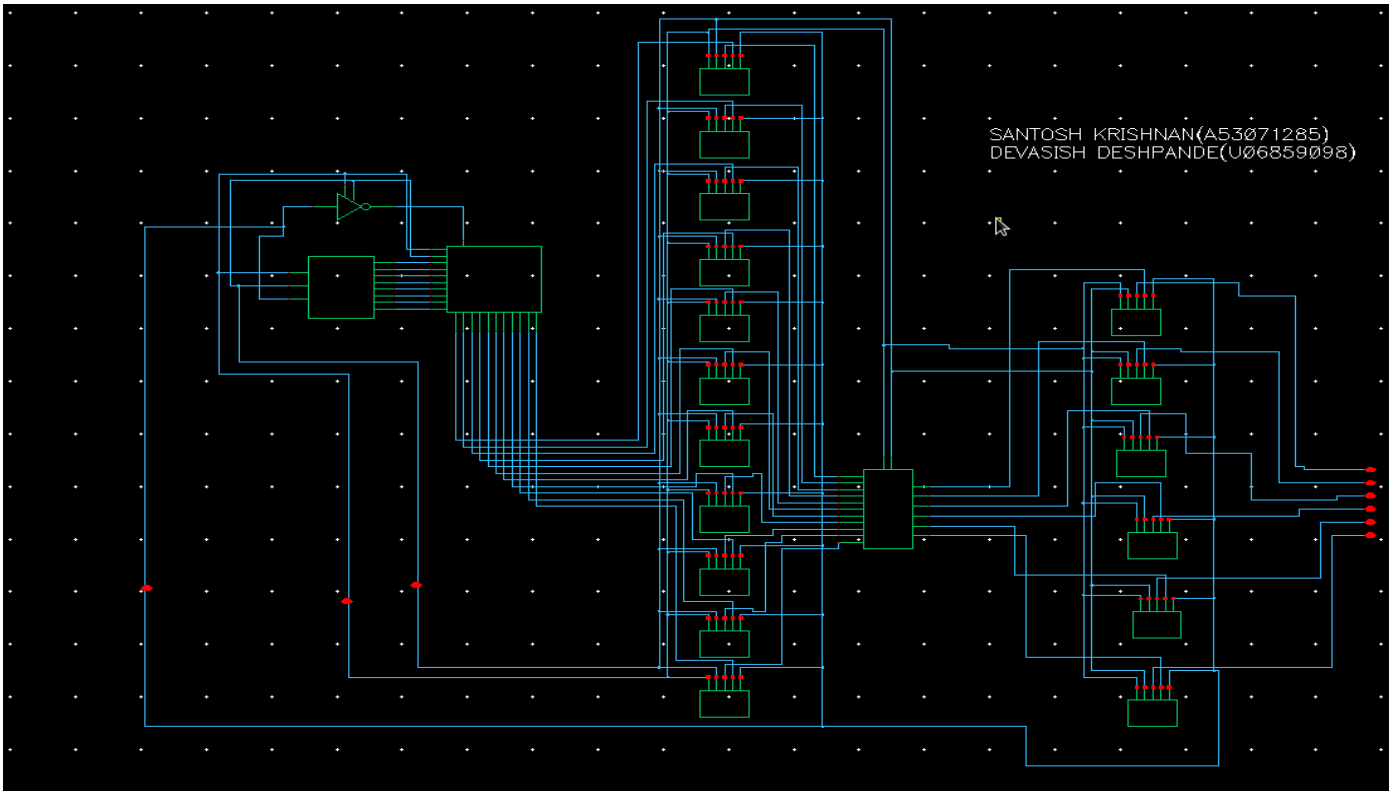
WHAT DIDN'T GIVE EXPETED RESULTS

The above mentioned techniques helped us save quite a substantial amount of energy. However there were some techniques that did not give us the savings energy that we expected. One such technique was pipelining. Prior to pipelining the lowest operational VDD was 560mV. However after pipelining we were able to reduce the VDD to only 520mV. The main reason for this was that the critical path of the second stage was much larger than that of the first stage. This problem can be avoided by either pipelining the ALU or by usng the concept of time borrowing. The other technique that didn't work as expected was the use of power gating. We initially planned to use power gating for all the sub blocks in the ALU. However on incorporating the feature in our design we found out that there was no reduction in power. On the contrary, it slightly increased for some blocks like the multiplier. Also the power gate had to be of a very large size in order to provide sufficient current to the sub blocks. Switching such large transistors result in an increase in dynamic power

SUGGESTION FOR FUTURE ITERATIONS

The entire project was completed in two weeks. Substantial energy savings were achieved by the techniques mention in this paper. However one can achieve far greater savings by proper pipelining of our design to make the critical path of all stages almost same. As mentioned above, we can also try to pipeline our ALU and use concepts of time borrowing to make the critical paths almost equal. Another technique that needs further investigation is the concept of power gating and the sizing of the header switch transistor. This technique is very useful if the circuit remains inactive for long periods. Our analysis shows us that if the adder-cum-subtraction block and multiplier remain inactive for a period of 1us and is then active for 10ns then the saving in power obtained is nearly 50% each.

SCHEMATIC OF SINGLE CYCLE ROCESSOR

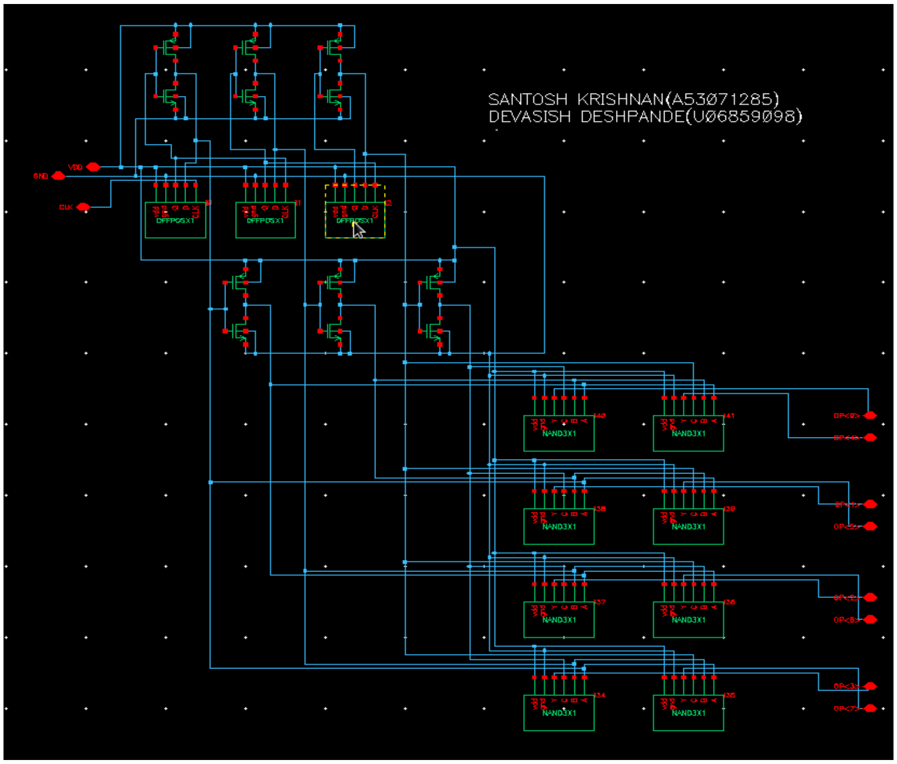


RESULT TABLE

INSTRUCTION	EXPECTED RESULTS
NOP 0000 0101	000000
Add 1101 1111	011100
And 0111 0101	000101
Sub 1011 0101	000110
Or 0001 1010	001011
Multi 0111 0111	110001
Slt 1000 1100	000001
NOP 0111 0011	000000

Figure 1 | SCHEMATIC OF SINGLE CYCLE PROCESSOR AND RESULT TABLE

Pointer counter and Address decoder



ROM

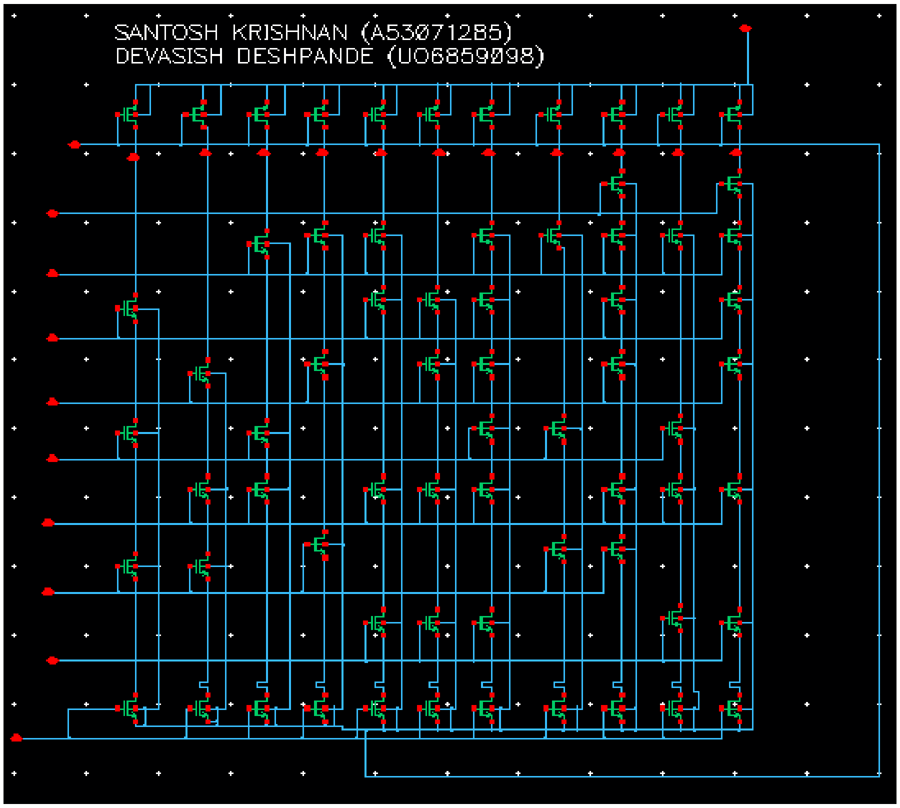


Figure 2 | SCHEMATIC OF POINTER COUNTER ROM ADDRESS DECODER

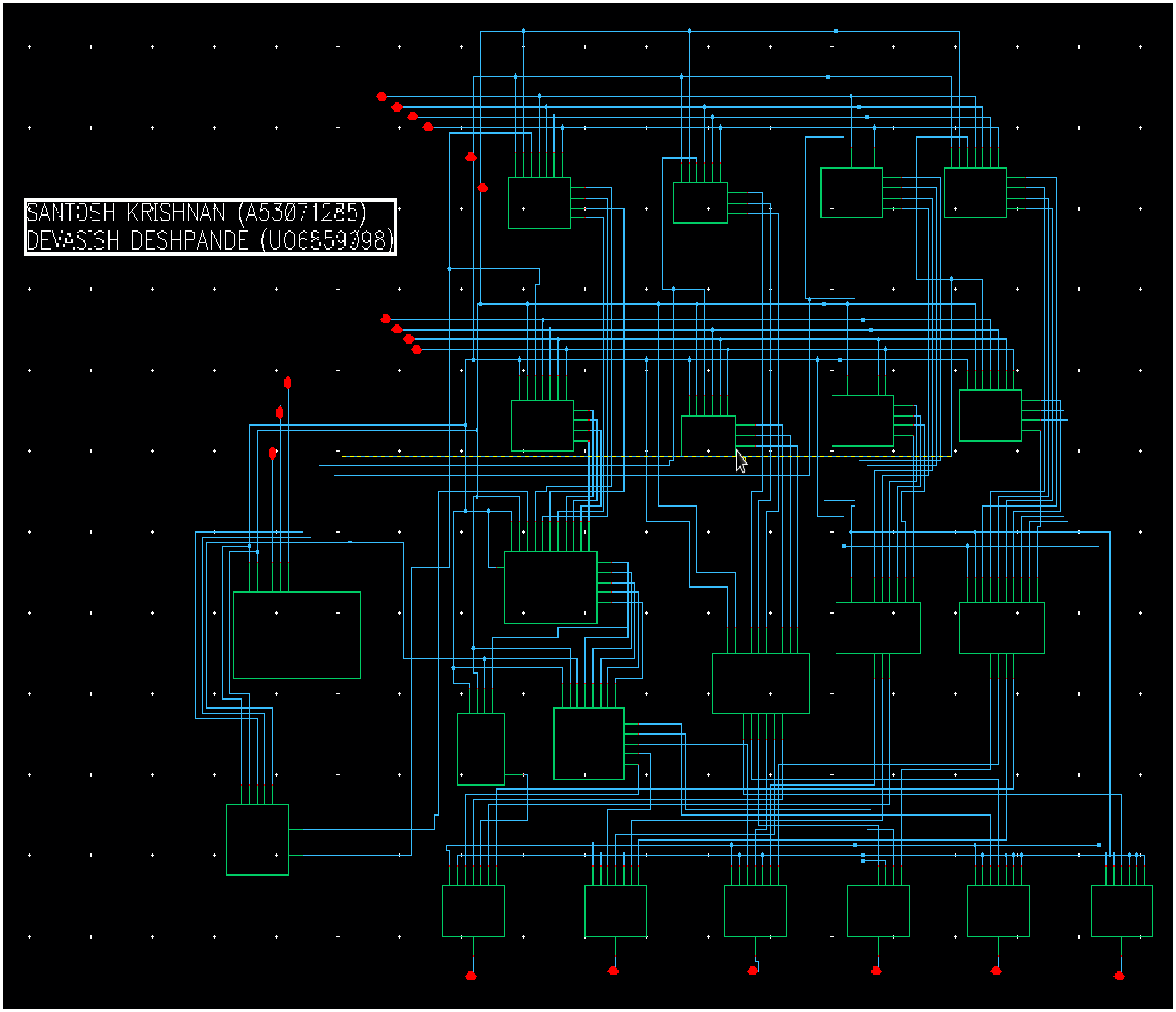
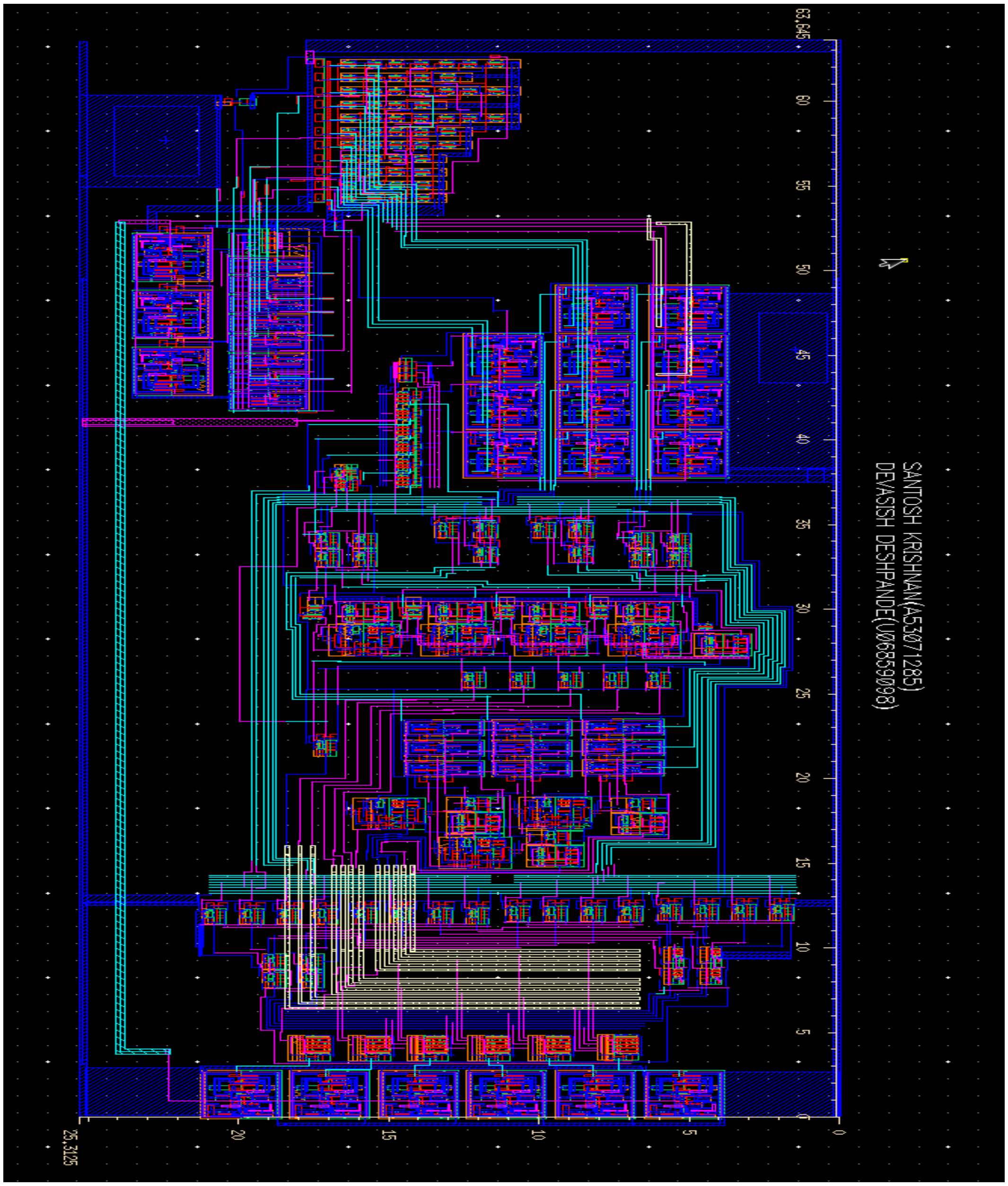
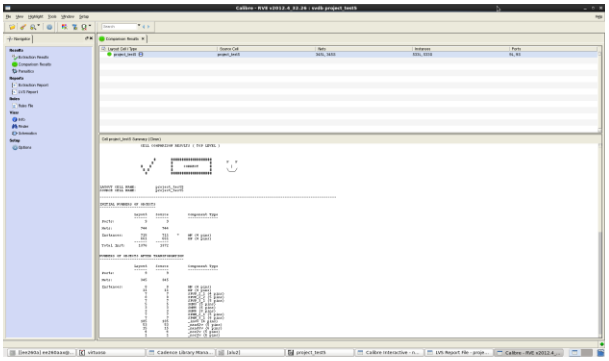


Figure 3 | SCHEMATIC OF ALU



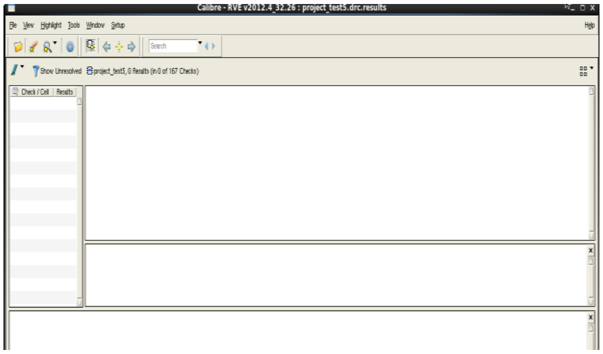

Calibre - RVE v2012.4_32.26 : svdb project_test5

Cell project_test5 Summary (Clean)

CELL COMPARISON RESULTS (TOP LEVEL)

CORRECT

LVS

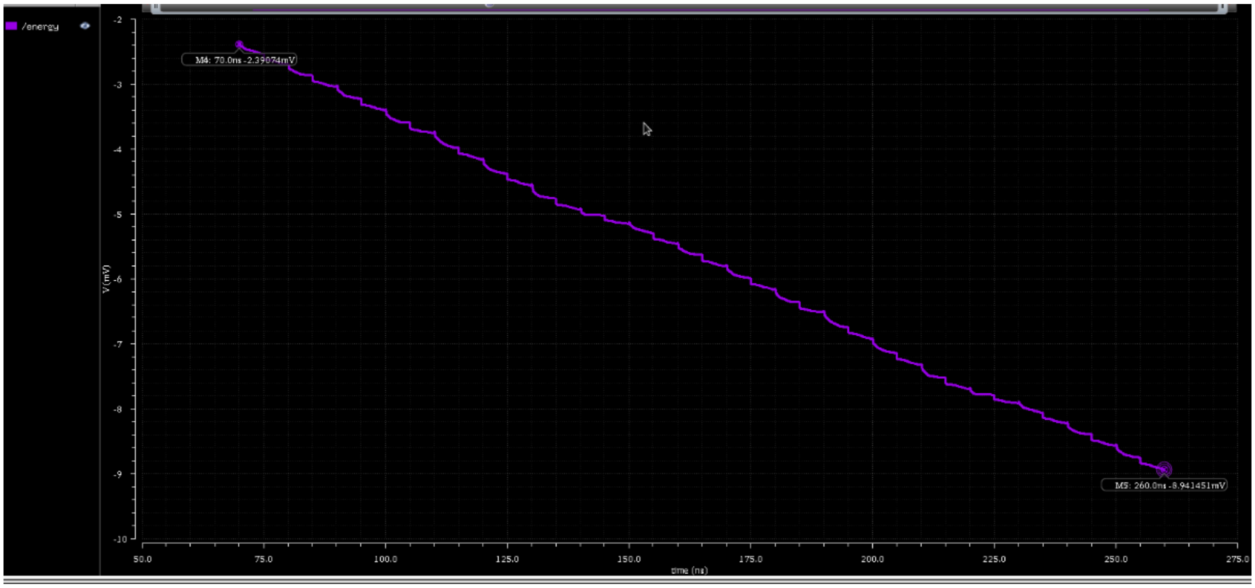

Calibre - RVE v2012.4_32.26 : project_test5.drc.results

project_test5, 0 Results (in 0 of 167 Checks)

DRC

Figure 4 | LAYOUT OF SINGLE CYCLE PROCESSOR , LVS AND DRC CONFORMATION

Vcap.



Simulation results post layout

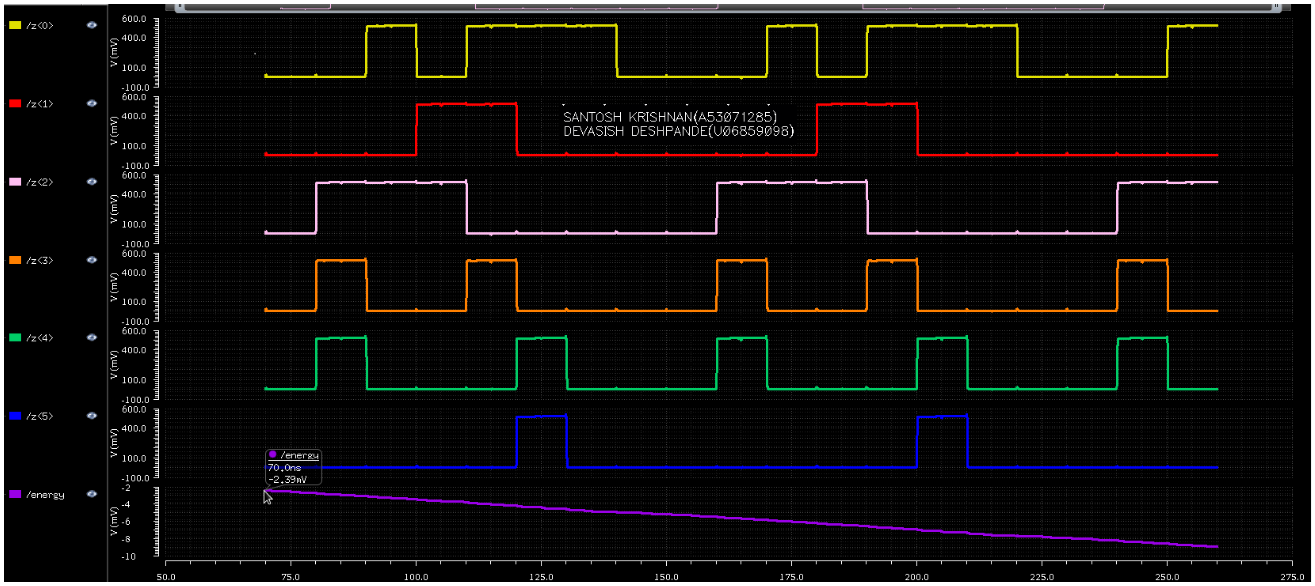


Figure 5 |TRANSIENT ANALYSIS

PARAMETER	SCHEMATIC SIMULATIONS	POST LAYOUT SIMULATIONS
Supply Voltage (v)	0.49	0.52
Logic style	Dynamic logic, Static CMOS	Dynamic Logic, Static CMOS
Clock Frequency (MHz)	100	100
Total energy consumed (nJ)	1.5930	3.406
Average energy per operation (nJ)	0.083934	0.17927
Average Power Consumption (uW)	8.3934	17.9270
Core Area (um^2)	1611.0	1611.0
Other Important information	E_{proc}= $V_{cap}V_{DD}C$ C=1nF V_{cap}=3.2511mV V_{DD}=0.49V Find E_{proc}.Avg Energy per operation= E_{proc}(T/10ns) T=190ns(from 70ns to 260ns) Avg Power = E_{proc}/T	E_{proc}= $V_{cap}V_{DD}C$ C=1nF V_{cap}=3.08mV V_{DD}=0.49V Find E_{proc}.Avg Energy per operation= E_{proc}(T/10ns) T=190ns(from 70ns to 260ns) Avg Power = E_{proc}/T

Figure 6 | TABLE OF RESULTS

Optional Figure 7 | Insert caption here