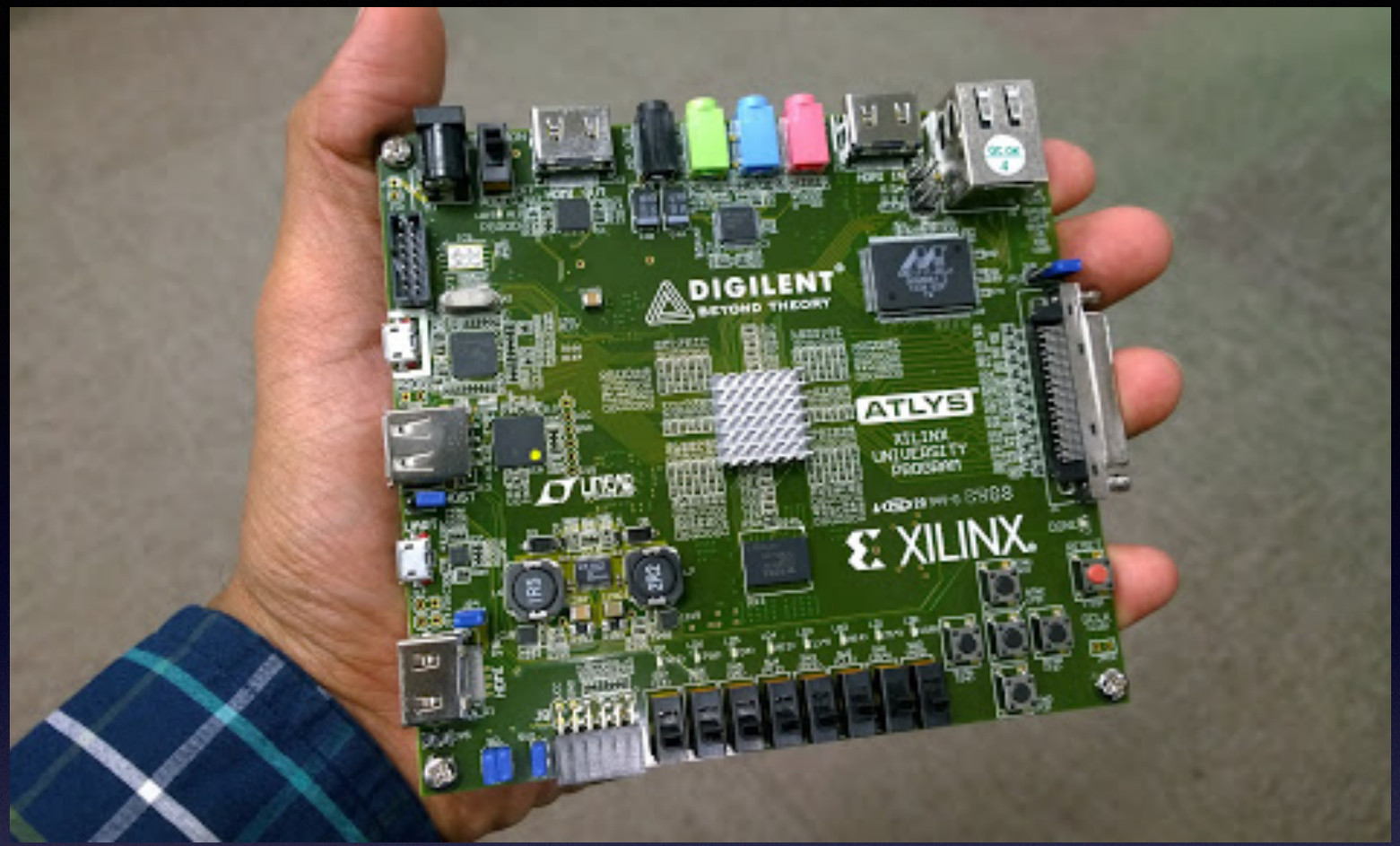


MIPS Processor

Group- A03

Dharmik Bhayani - 201501010
Jaymeen Kachrola - 201501042
Jenil Kansara - 201501044

- A MIPS processor is one version of a reduced instruction set computer (RISC). It is a model that is studied often in university computer architecture courses because it has enough complexity to include all the main aspects of modern processors, but isn’t overly complex so as for us it is easy to learn it in the details.

- MIPS processors have been around since the early 1980’s and gained much popularity in the 1990’s as it was estimated that one third of all RISC processors used the MIPS architecture, including embedded systems for the Sony PlayStation 2 and the PlayStation Portable.

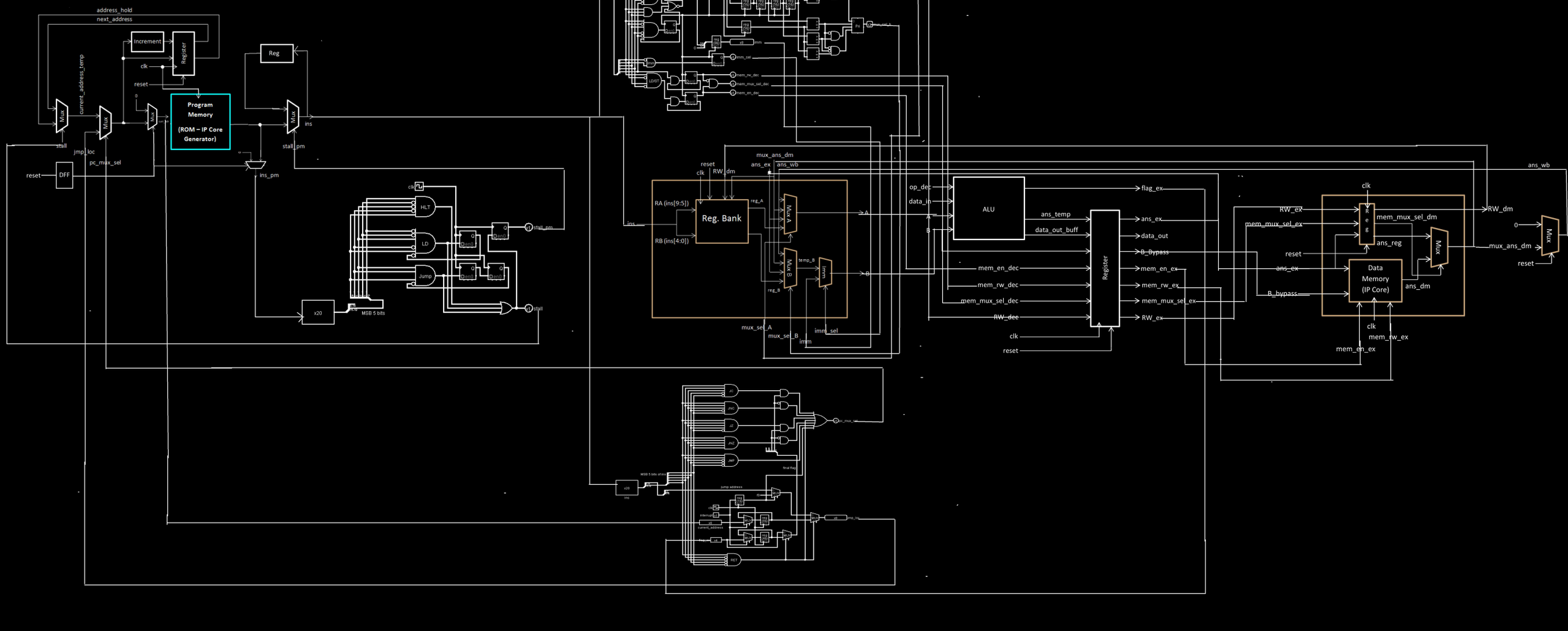
Motivation

- To learn how to implement all the aspects of the processor simultaneously.
- To work on pipelined structure, which is used everywhere.
- To learn how to implement IP-Core for memory uses in Xilinx.
- To learn to use post-route simulation in Xilinx.
- Finally, also for the experience of having made a working processor.

Specifications

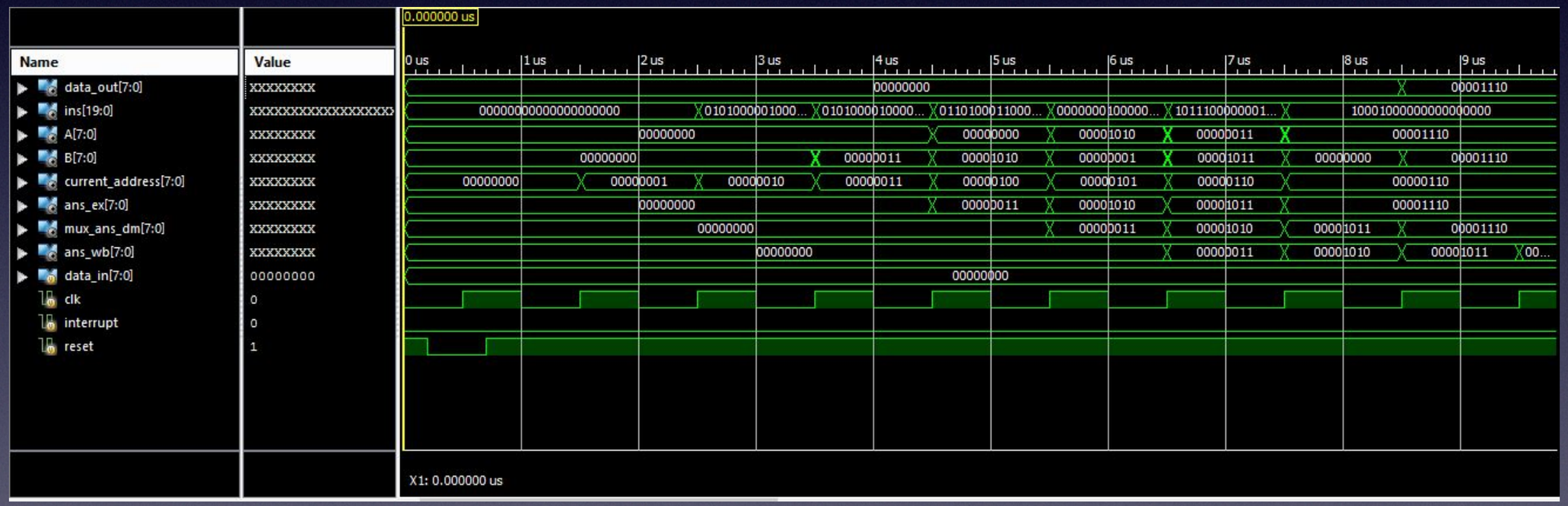
8 Bit Processor
5 Bit opcode
RISC based architecture
28 different opcodes
8 X 32 bit Register file
8 X 256 bits of Data Memory
3 forwarding units (for preventing Hazards)
clock cycle - 1 micro second (frequency - 1 Megahertz)
3.5 clock cycles per Instruction
5 stage pipeline
Stage 1 : Instruction Fetch
Stage 2 : Instruction Decode
Stage 3 : Execution
Stage 4 : Data Memory
Stage 5 : Write Back

Data Path


address_hold
next_address
Increment
Register
clk
reset
current_address_temp
Mux
Mux
Mux
Mux
stall
jmp_loc
pc_mux_sel
Program Memory (ROM – IP Core Generator)
ins
stall_pm
Reg
ins_pm
DFF
reset
clk
HLT
LD
Jump
stall_pm
stall
x20
MSB 5 bits
op_dec
data_in
A
B
ALU
ans_temp
data_out_buff
flag_ex
ans_ex
data_out
B_Bypass
mem_en_ex
mem_rw_ex
mem_mux_sel_ex
RW_ex
mem_en_dec
mem_rw_dec
mem_mux_sel_dec
RW_dec
clk
reset
Register
mux_ans_dm
ans_ex
ans_wb
reset
clk
RW_dm
RA (ins[9:5])
RB (ins[4:0])
Reg. Bank
reg_A
reg_B
Mux A
Mux B
temp_B
imm
mux_sel_A
mux_sel_B
imm_sel
imm
clk
RW_ex
mem_mux_sel_ex
reset
ans_ex
B_bypass
Data Memory (IP Core)
mem_mux_sel_dm
ans_reg
ans_dm
Mux
clk
mem_rw_ex
mem_en_ex
RW_dm
0
mux_ans_dm
Mux
reset
ans_wb
JC
JNC
JZ
JNZ
JMP
MSB 5 bits of ins
ins
jump address
final flag
internal
current_address
flag_ex
RST
pc_mux_sel

Simulation Results

Addition : We got the output at 8.506404 micro seconds



Multiplication : We got the output at 38.507048 micro seconds

Comparison with 8085 processor

- This processor is faster than 8085.
- CPI count is larger than 8085 microprocessor.
- Clock frequency of 8085 can be 3,5 or 6 Megahertz . This processor's is 1 Megahertz
- Opcode is 8 bit in 8085 processor , when this processor's is 5

Challenges faced during design and implementation

- Minimum use of behavioural modelling.
- We were not getting the output at desired time.
- We were not stalling the clock cycles by sufficient amount for conditional jumps.

Feedback

- We are now aware of how the building of processors actually work in the real world industries after taking this subject.
- We completed all the Lab Sessions, they all were quite active and the TA's were helpful we hardly faced any technical problems during the lab.