Digital Engineering
Lab 3

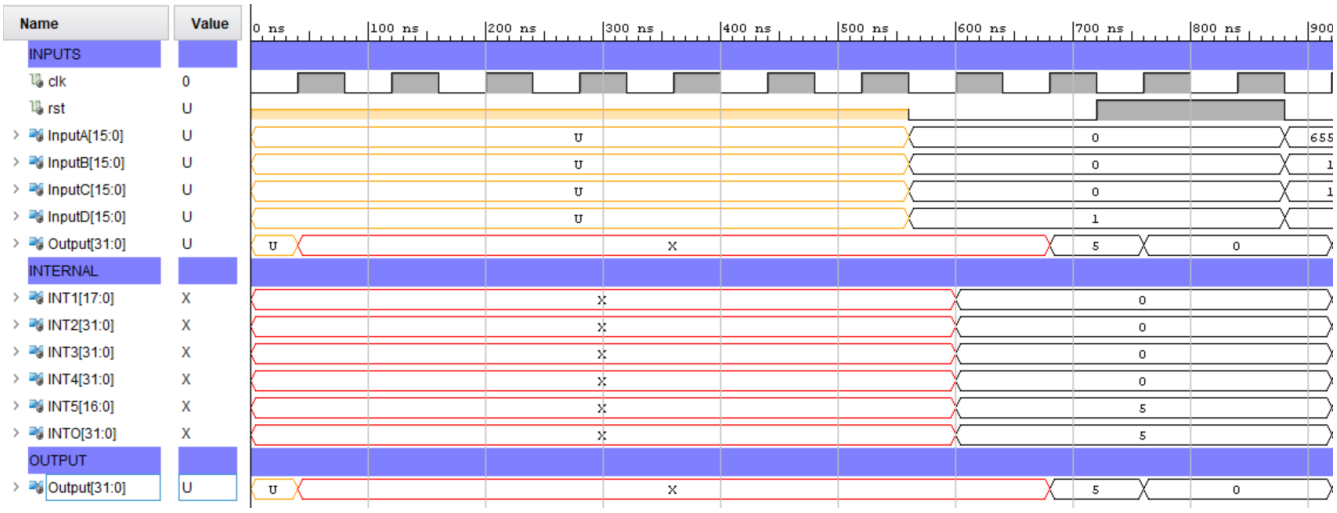
Y3890959
Y3878784

7th February 2023

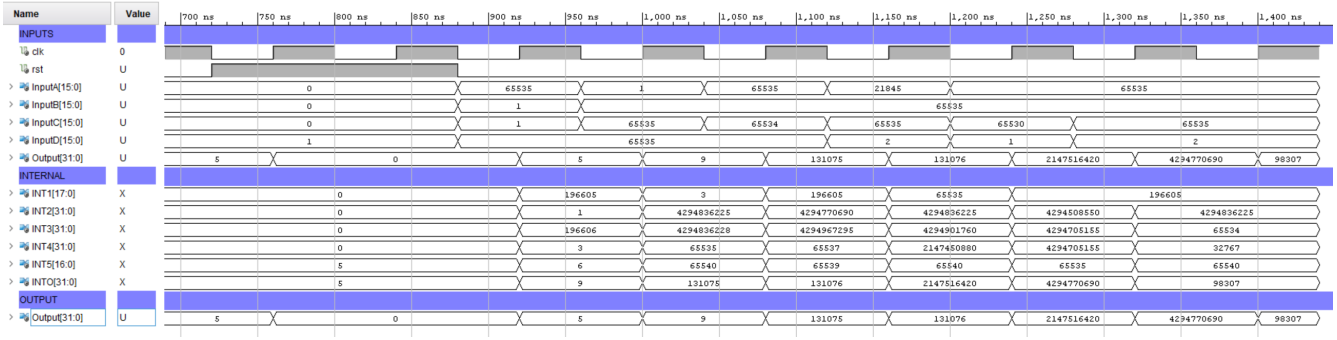
Task A: Pipelining

2.1.1 Behaviour Simulation

Waveform 1: Global Initialisation/Reset



Waveform 2: Test Sequence



Console Output

```
restart
INFO: [Simtcl 6-17] Simulation restarted
run 6 us
Note: TEST VECTOR 0 PASS.
Time: 1040 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3/Lab3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 1 PASS.
Time: 1120 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3/Lab3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 2 PASS.
Time: 1200 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3/Lab3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 3 PASS.
Time: 1280 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3/Lab3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 4 PASS.
Time: 1360 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3/Lab3.srcs/sim_1/imports/new/algorithm_tb.vhd
Failure: TEST VECTOR 5 A = 65535 B = 65535 C = 65535 D = 2 O_observed = 98307 O_expected = -2147385341
Time: 1440 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3/Lab3.srcs/sim_1/imports/new/algorithm_tb.vhd
$finish called at time : 1440 ns : File "C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3/Lab3.srcs/sim_1/imports/new/algorithm_tb.vhd" Line 173
```

2.1.2 Design Runs

Tcl Console | Messages | Log | Reports | **Design Runs**

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
✓ synth_1	constrs_1	synth_design Complete!								897	96	0.00	0	0
✓ impl_1	constrs_1	route_design Complete!	0.383	0.000	0.532	0.000	0.000	0.109	0	886	96	0.00	0	0

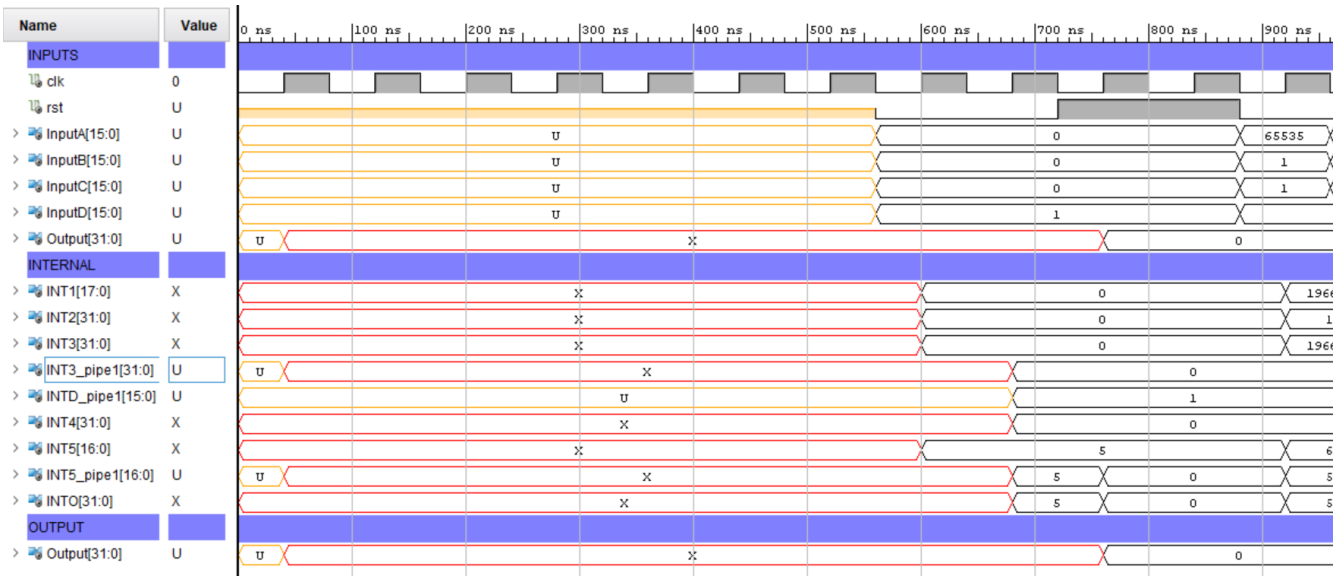
The best period where the constraints are met is at 107ns with a WNS of 0.383ns. The tools fail with a 106ns constraint. The fastest frequency at which this design can run, taking into account the WNS value, is 9.379MHz.

2.1.3 Post-Route Timing Report: Max Delay Path

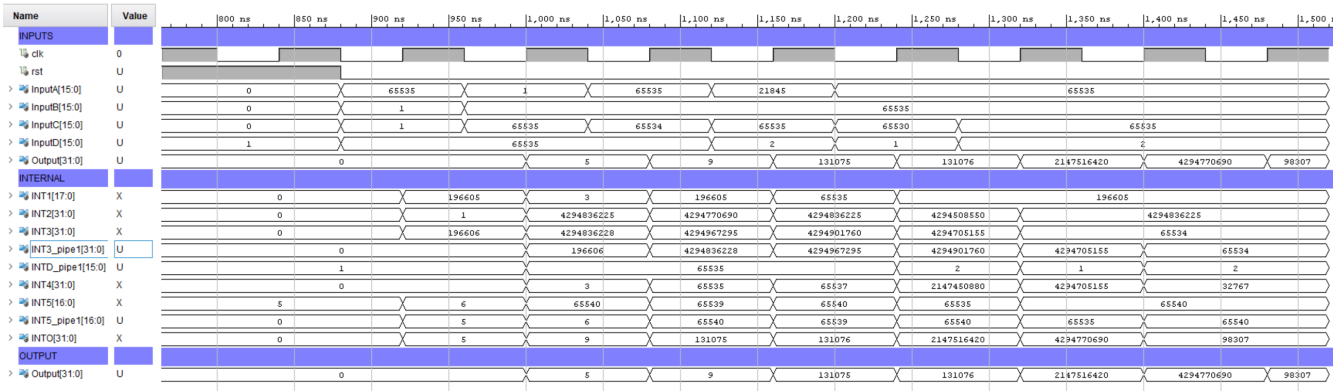
```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.383ns  (required time - arrival time)
  Source:                 INTB_reg[11]/C
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@53.500ns period=107.000ns})
  Destination:            O_reg[29]/D
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@53.500ns period=107.000ns})
  Path Group:             clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            107.000ns  (clk rise@107.000ns - clk rise@0.000ns)
  Data Path Delay:        106.462ns  (logic 49.185ns (46.200%)  route 57.277ns (53.800%))
```

2.1.4 Pipeline 1 Behavioural Simulation

Waveform 1: Global Initialisation/Reset



Waveform 2: Test Sequence



Console

```
restart
INFO: [Simtcl 6-17] Simulation restarted
run 6 us
Note: TEST VECTOR 0 PASS.
Time: 1120 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]Documents/GitHub/DigitalEngineering2023/Lab3P1/Lab3P1.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 1 PASS.
Time: 1200 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]Documents/GitHub/DigitalEngineering2023/Lab3P1/Lab3P1.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 2 PASS.
Time: 1280 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P1/Lab3P1.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 3 PASS.
Time: 1360 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P1/Lab3P1.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 4 PASS.
Time: 1440 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P1/Lab3P1.srcs/sim_1/imports/new/algorithm_tb.vhd
Failure: TEST VECTOR 5 A = 65535 B = 65535 C = 65535 D = 2 O_observed = 98307 O_expected = -2147385341
Time: 1520 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P1/Lab3P1.srcs/sim_1/imports/new/algorithm_tb.vhd
$finish called at time : 1520 ns : File "C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P1/Lab3P1.srcs/sim_1/imports/new/algorithm_tb.vhd" Line 173
```

2.1.5 Design Runs

Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
synth_1	constrs_1	synth_design Complete!								897	161	0.00	0	0
impl_1	constrs_1	route_design Complete!	0.431	0.000	0.143	0.000	0.000	0.110	0	887	161	0.00	0	0

The best period where the constraints are met is at 96ns with a WNS of 0.431ns. The tools fail with a 95ns constraint. The fastest frequency at which this design can run, taking into account the WNS value, is 10.37MHz.

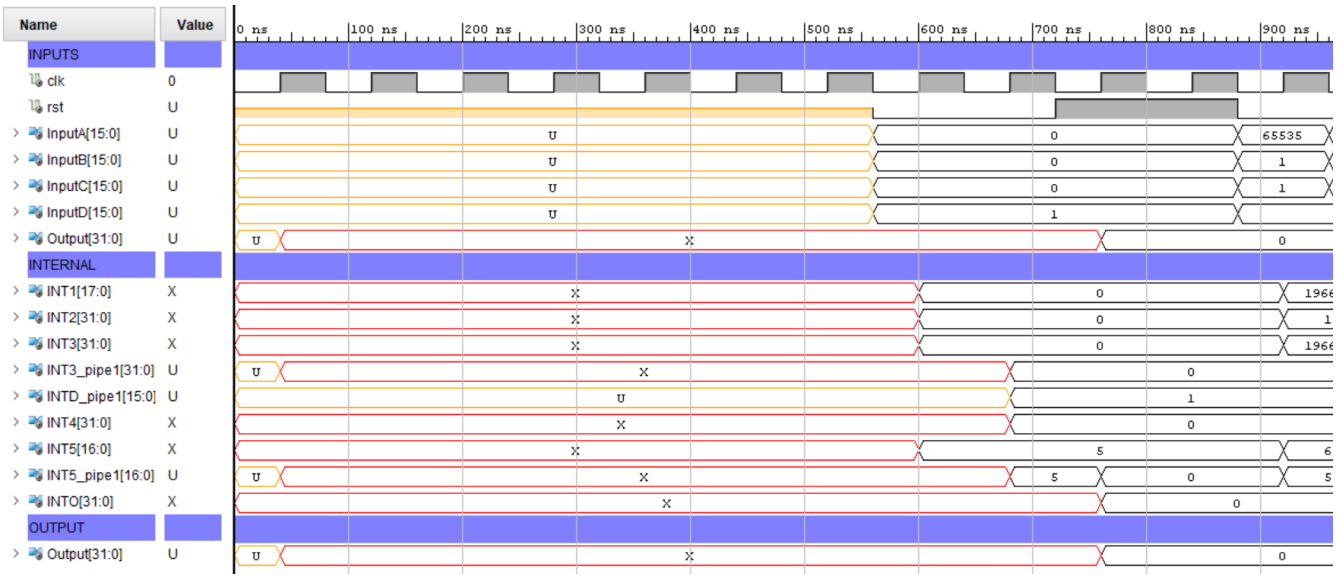
2.1.6 Post-Route Timing Report: Max Delay Path

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.431ns  (required time - arrival time)
  Source:                 INTD_pipe1_reg[10]/C
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@48.000ns period=96.000ns})
  Destination:            O_reg[29]/D
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@48.000ns period=96.000ns})
  Path Group:             clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            96.000ns  (clk rise@96.000ns - clk rise@0.000ns)
  Data Path Delay:        95.395ns  (logic 44.272ns (46.409%)  route 51.123ns (53.591%))
```

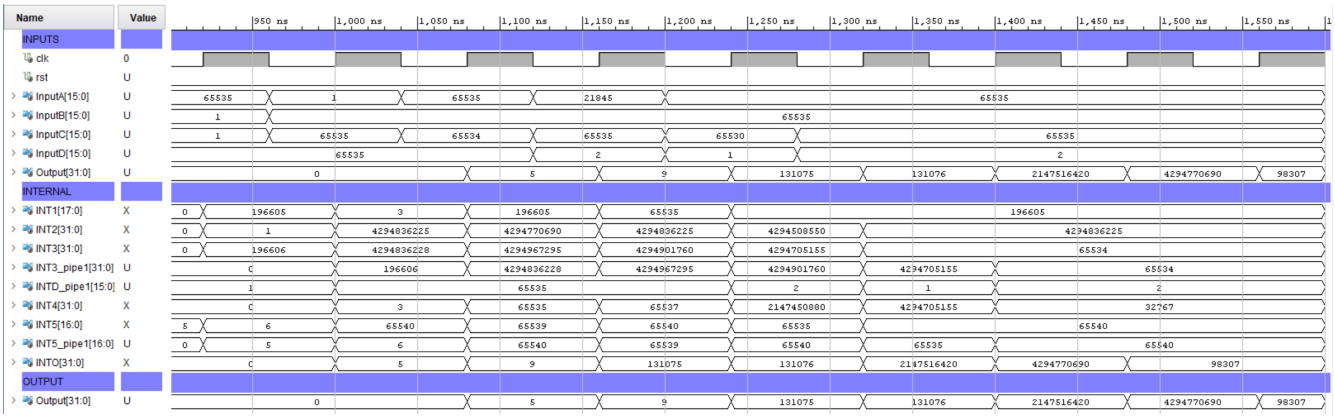
Before the pipeline was added, the critical path was detected between INTB_reg and O_reg. Now, we've reduced the effect of that by adding a pipeline. Now the detected critical path is between INTD_pipe1_reg and O_reg. The critical path consists of the division and the addition operation, This does in fact increase the max clock period that the circuit can support as per the theory at the cost of reducing latency. The clock period we achieved in 2.1.2 was 107ns, with a pipeline stage, we can achieve 96ns.

2.1.7 Pipeline 2 Behavioural Simulation

Waveform 1: Global Initialisation/Reset



Waveform 2: Test Sequence



Console

```
restart
INFO: [Simtcl 6-17] Simulation restarted
run 6 us
Note: TEST VECTOR 0 PASS.
Time: 1200 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P2/Lab3P2.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 1 PASS.
Time: 1280 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P2/Lab3P2.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 2 PASS.
Time: 1360 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P2/Lab3P2.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 3 PASS.
Time: 1440 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P2/Lab3P2.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 4 PASS.
Time: 1520 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P2/Lab3P2.srcs/sim_1/imports/new/algorithm_tb.vhd
Failure: TEST VECTOR 5 A = 65535 B = 65535 C = 65535 D = 2 O_observed = 98307 O_expected = -2147385341
Time: 1600 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P2/Lab3P2.srcs/sim_1/imports/new/algorithm_tb.vhd
$finish called at time : 1600 ns : File "C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P2/Lab3P2.srcs/sim_1/imports/new/algorithm_tb.vhd" Line 173
```

2.1.8 Design Runs

Tcl Console | Messages | Log | Reports | **Design Runs**

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
synth_1	constrs_1	synth_design Complete!								897	210	0.00	0	0
impl_1	constrs_1	route_design Complete!	0.574	0.000	0.076	0.000	0.000	0.110	0	887	210	0.00	0	0

The best period where the constraints are met is at 90ns with a WNS of 0.574ns. The tools fail with an 89ns constraint. The fastest frequency at which this design can run, taking into account the WNS value, is 11.18MHz.

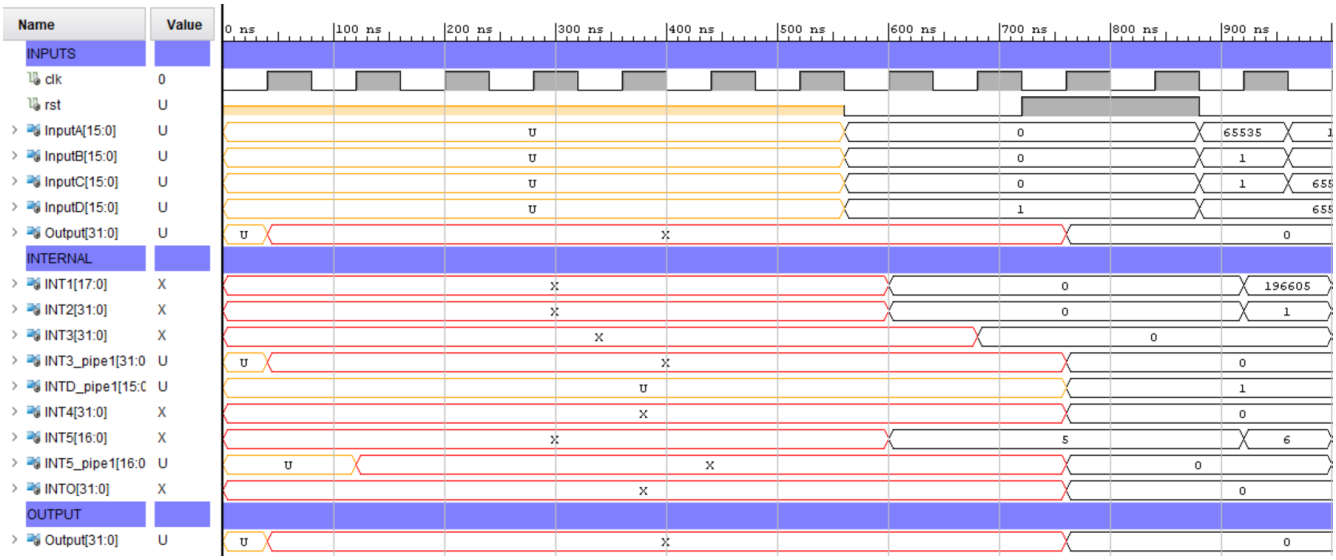
2.1.9 Post-Route Timing Report: Max Delay Path

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.574ns  (required time - arrival time)
  Source:                 INTD_pipe1_reg[5]/C
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@45.000ns period=90.000ns})
  Destination:            INT4_pipe2_reg[0]/D
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@45.000ns period=90.000ns})
  Path Group:             clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            90.000ns  (clk rise@90.000ns - clk rise@0.000ns)
  Data Path Delay:        89.373ns  (logic 43.001ns (48.114%)  route 46.372ns (51.886%))
```

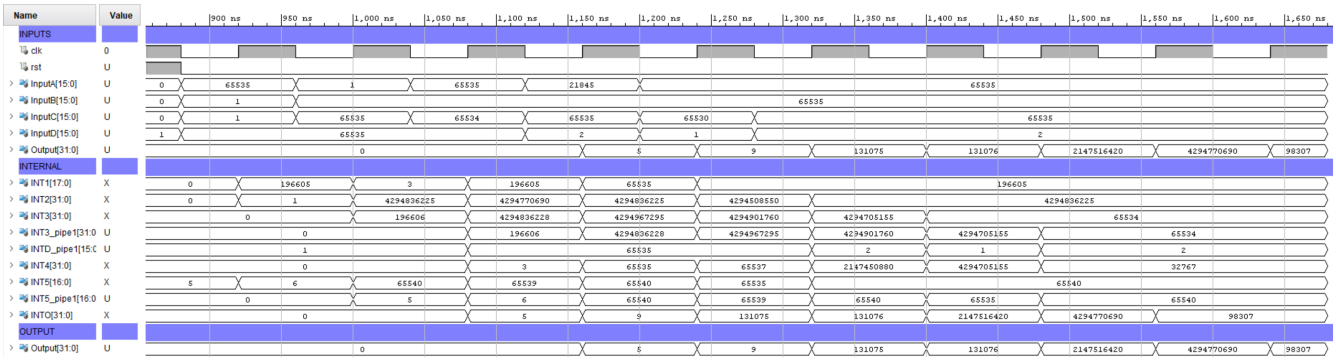
After pipeline 1 was added, the critical path was detected between INTD_reg and O_reg. Now, we've reduced the effect of that by adding pipeline 2. Now the detected critical path is between INTD_pipe1_reg and INT4_pipe2_reg. The critical path consists of just the division operation, This does in fact increase the max clock period that the circuit can support as per the theory at the cost of reducing latency. The clock period we achieved in 2.1.2 was 96ns, with a pipeline stage, we can achieve 90ns.

2.1.10 Behavioural Simulation

Waveform 1: Global Initialisation/Reset



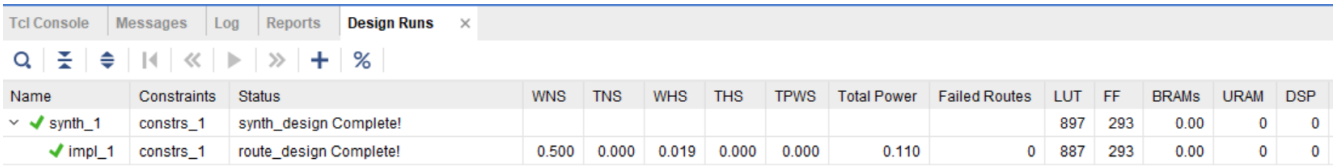
Waveform 2: Test Sequence



Console

```
restart
INFO: [Simtcl 6-17] Simulation restarted
run 6 us
Note: TEST VECTOR 0 PASS.
Time: 1280 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P3/Lab3P3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 1 PASS.
Time: 1360 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P3/Lab3P3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 2 PASS.
Time: 1440 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P3/Lab3P3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 3 PASS.
Time: 1520 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P3/Lab3P3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 4 PASS.
Time: 1600 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P3/Lab3P3.srcs/sim_1/imports/new/algorithm_tb.vhd
Failure: TEST VECTOR 5 A = 65535 B = 65535 C = 65535 D = 2 O_observed = 98307 O_expected = -2147385341
Time: 1680 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P3/Lab3P3.srcs/sim_1/imports/new/algorithm_tb.vhd
$finish called at time : 1680 ns : File "C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3P3/Lab3P3.srcs/sim_1/imports/new/algorithm_tb.vhd" Line 173
```

2.1.11 Design Runs



Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
synth_1	constrs_1	synth_design Complete!								897	293	0.00	0	0
impl_1	constrs_1	route_design Complete!	0.500	0.000	0.019	0.000	0.000	0.110	0	887	293	0.00	0	0

The best period where the constraints are met is at 90ns with a WNS of 0.5ns. The tools fail with an 89ns constraint. The fastest frequency at which this design can run, taking into account the WNS value, is 11.17MHz.

2.1.12 Post-Route Timing Report: Max Delay Path

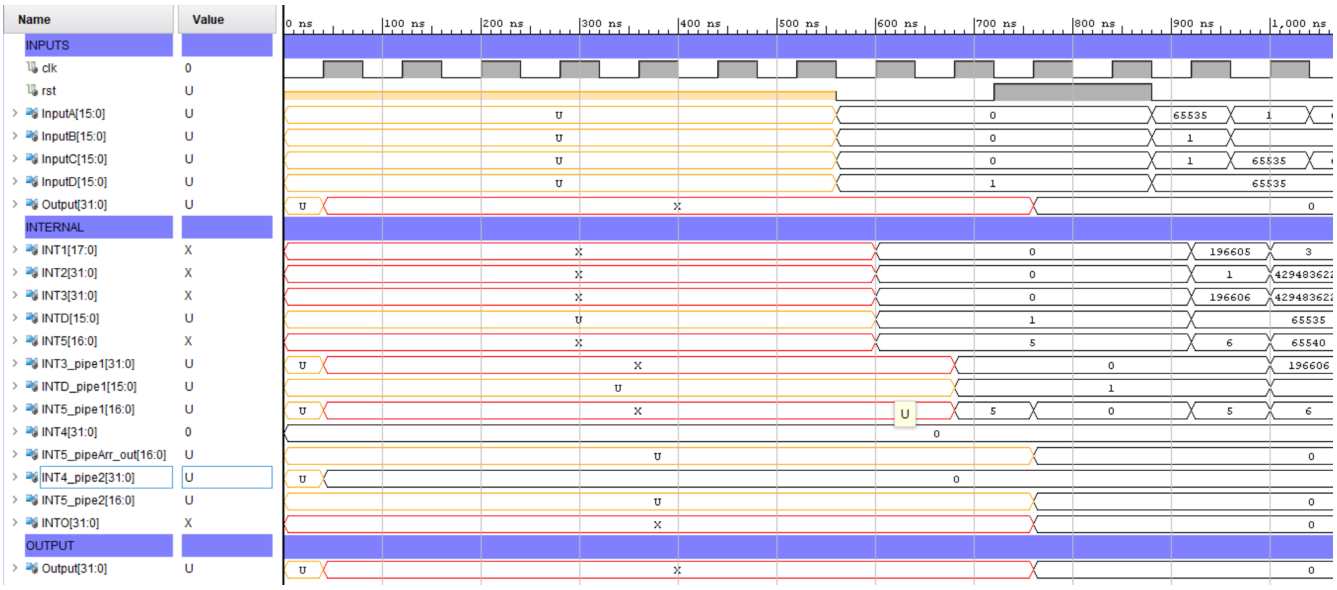
```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.500ns  (required time - arrival time)
  Source:                 INTD_pipe1_reg[9]/C
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@45.000ns period=90.000ns})
  Destination:            INT4_pipe2_reg[0]/D
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@45.000ns period=90.000ns})
  Path Group:             clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            90.000ns  (clk rise@90.000ns - clk rise@0.000ns)
  Data Path Delay:        89.535ns  (logic 42.507ns (47.475%)  route 47.028ns (52.525%))
```

After pipeline 2 was added, the critical path was detected between INTD_pipe1_reg and INT4_pipe2_reg. Since adding the third pipeline, we don't see an improvement, but rather we see a decline in performance. The clock period we achieved in 2.1.2 was 90ns, with this pipeline, the clock period hasn't changed.

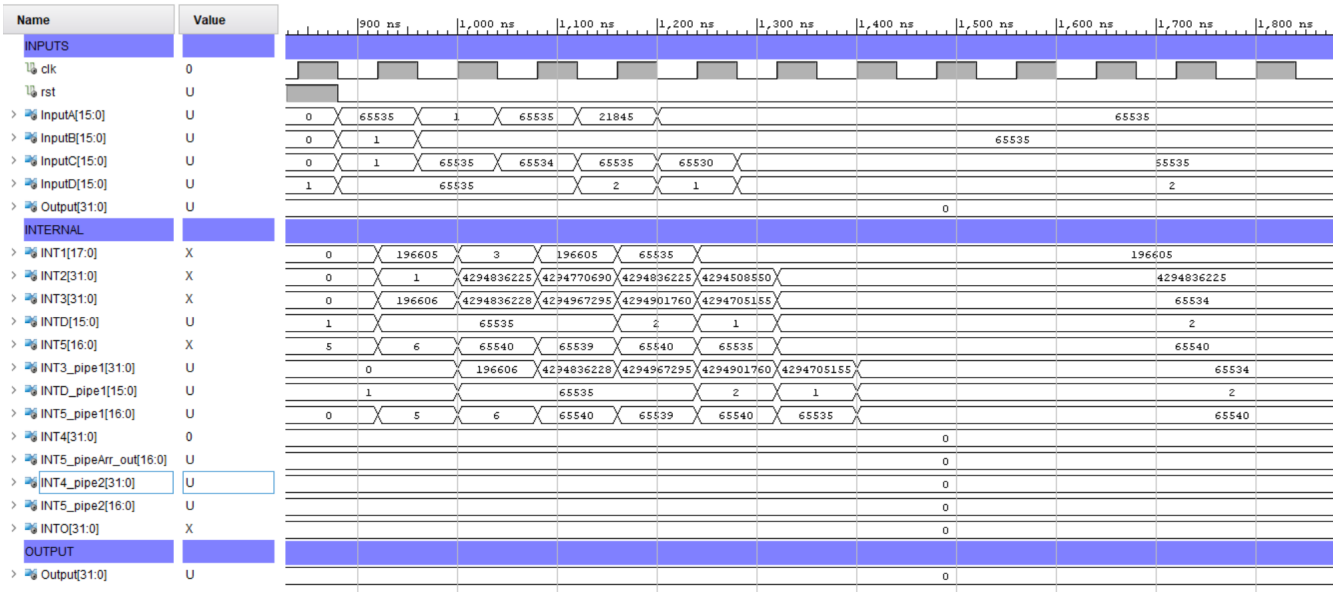
Task B: IP Components

2.2.1 Behavioural Simulation

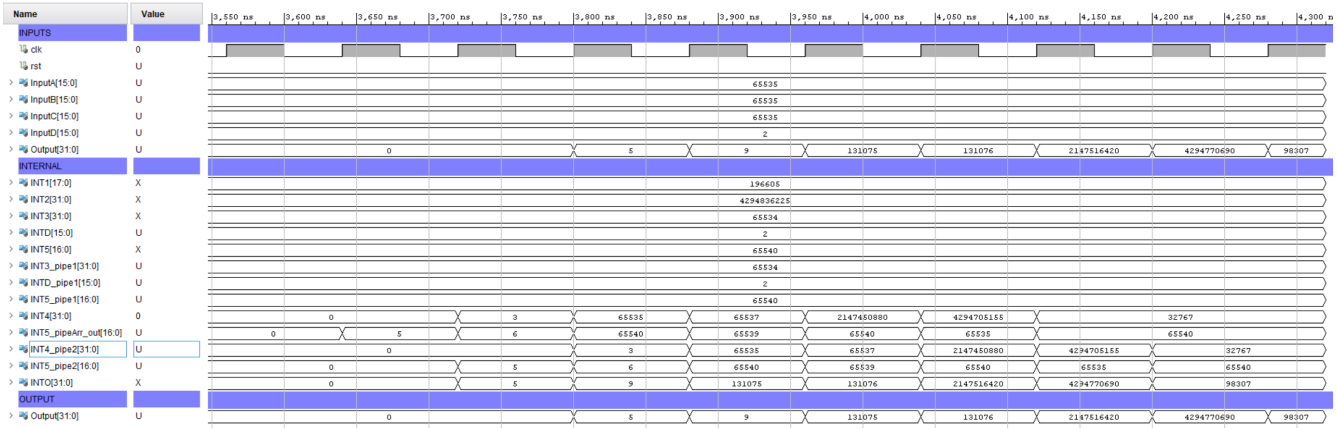
Waveform 1: Global Initialisation/Reset



Waveform 2: Test Sequence 1



Waveform 3: Test Sequence 2



Console

```
restart
INFO: [Simtcl 6-17] Simulation restarted
run 6 us
Note: TEST VECTOR 0 PASS.
Time: 3920 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3TaskB/Lab3TaskB.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 1 PASS.
Time: 4 us  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3TaskB/Lab3TaskB.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 2 PASS.
Time: 4080 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3TaskB/Lab3TaskB.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 3 PASS.
Time: 4160 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3TaskB/Lab3TaskB.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 4 PASS.
Time: 4240 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3TaskB/Lab3TaskB.srcs/sim_1/imports/new/algorithm_tb.vhd
Failure: TEST VECTOR 5 A = 65535 B = 65535 C = 65535 D = 2 O_observed = 98307 O_expected = -2147385341
Time: 4320 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3TaskB/Lab3TaskB.srcs/sim_1/imports/new/algorithm_tb.vhd
$finish called at time : 4320 ns : File "C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3TaskB/Lab3TaskB.srcs/sim_1/imports/new/algorithm_tb.vhd" Line 173
save_wave_config {C:/Users/[redacted]/Documents/GitHub/DigitalEngineering2023/Lab3TaskB/Lab3TaskB.srcs/sim_1/imports/Lab3/algorithm_tb_behav.wcfg}
```

2.2.2 Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
✓ synth_1	constrs_1	synth_design Complete!								1019	2423	0.00	0	0
✓ impl_1	constrs_1	route_design Complete, Failed Timing!	0.601	0.000	-0.075	-0.349	0.000	0.151	0	990	2341	0.00	0	0

The best period where the constraints are met is at 14ns with a WNS of 0.601ns (a massive improvement from the 90ns max period we achieved before). The tools fail with a 13ns constraint. The fastest frequency at which this design can run, considering the WNS value, is 74.63MHz. The vast increase in flip-flops has also been noted.

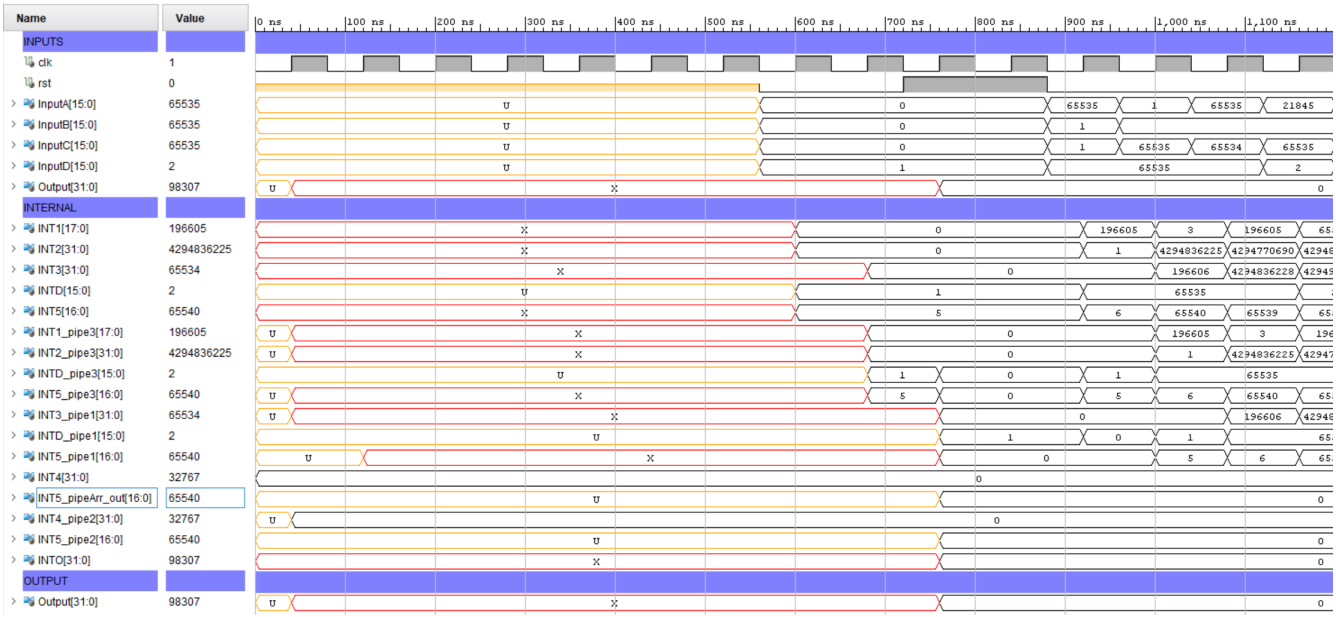
2.2.3 Post-Route Timing Report: Max Delay Path

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.601ns  (required time - arrival time)
  Source:                 INTB_reg[7]/C
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@7.000ns period=14.000ns})
  Destination:            INT3_pipe1_reg[29]/D
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@7.000ns period=14.000ns})
  Path Group:             clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            14.000ns  (clk rise@14.000ns - clk rise@0.000ns)
  Data Path Delay:        13.411ns  (logic 4.854ns (36.195%)  route 8.557ns (63.805%))
```

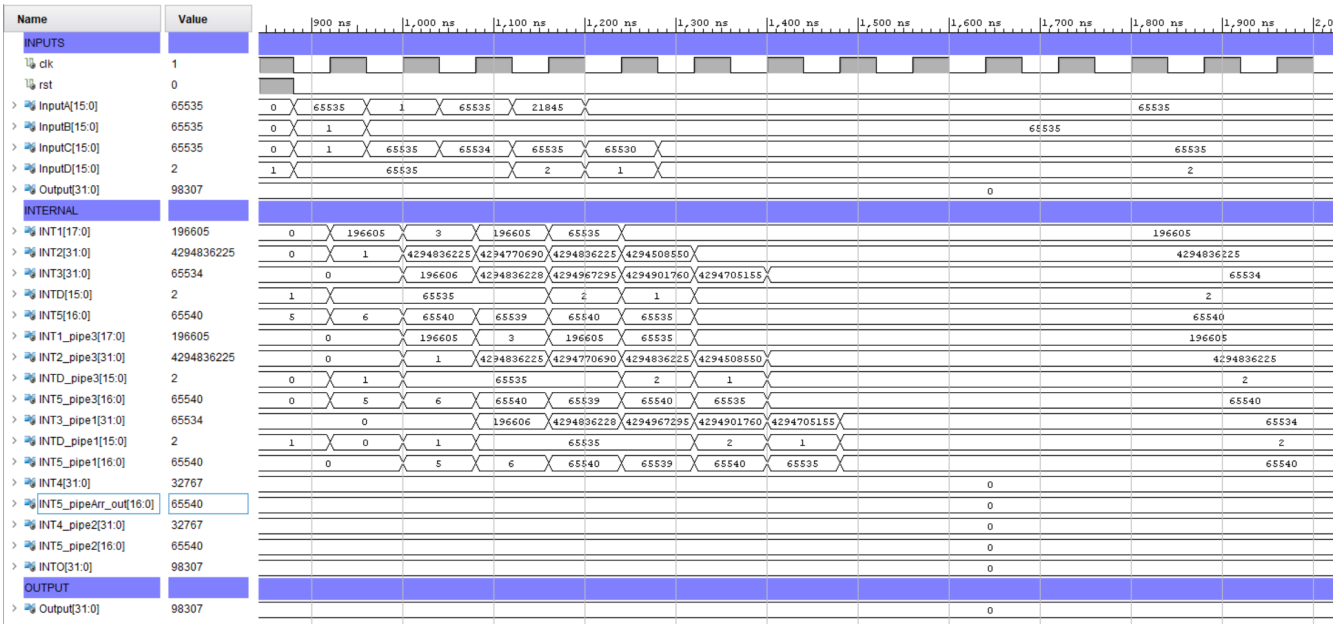
From this report, we can identify that the critical path lies between INTB_reg and INT3_pipe1_reg. The two mathematical operations which lie in the critical path is the multiplication and addition, this is before the first pipeline and after the input registers.

2.2.4 Behavioural Simulation

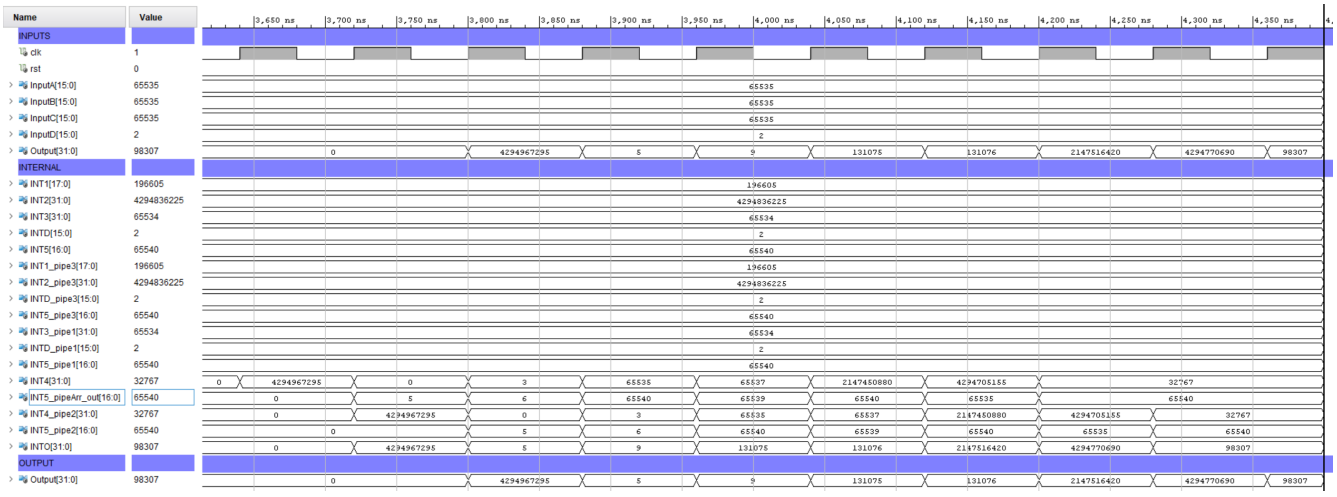
Waveform 1: Global Initialisation/Reset



Waveform 2: Test Sequence 1



Waveform 2: Test Sequence 2



Console

```
restart
INFO: [Simtcl 6-17] Simulation restarted
run 6 us
Note: TEST VECTOR 0 PASS.
Time: 4 us  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]Documents/GitHub/DigitalEngineering2023/Lab3TaskBP3/Lab3TaskBP3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 1 PASS.
Time: 4080 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]Documents/GitHub/DigitalEngineering2023/Lab3TaskBP3/Lab3TaskBP3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 2 PASS.
Time: 4160 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]Documents/GitHub/DigitalEngineering2023/Lab3TaskBP3/Lab3TaskBP3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 3 PASS.
Time: 4240 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]Documents/GitHub/DigitalEngineering2023/Lab3TaskBP3/Lab3TaskBP3.srcs/sim_1/imports/new/algorithm_tb.vhd
Note: TEST VECTOR 4 PASS.
Time: 4320 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]Documents/GitHub/DigitalEngineering2023/Lab3TaskBP3/Lab3TaskBP3.srcs/sim_1/imports/new/algorithm_tb.vhd
Failure: TEST VECTOR 5 A = 65535 B = 65535 C = 65535 D = 2 O_observed = 98307 O_expected = -2147385341
Time: 4400 ns  Iteration: 0  Process: /algorithm_tb/check_outputs  File: C:/Users/[redacted]Documents/GitHub/DigitalEngineering2023/Lab3TaskBP3/Lab3TaskBP3.srcs/sim_1/imports/new/algorithm_tb.vhd
$finish called at time : 4400 ns : File "C:/Users/[redacted]Documents/GitHub/DigitalEngineering2023/Lab3TaskBP3/Lab3TaskBP3.srcs/sim_1/imports/new/algorithm_tb.vhd" Line 173
save_wave_config {C:/Users/[redacted]Documents/GitHub/DigitalEngineering2023/Lab3TaskBP3/Lab3TaskBP3.srcs/sim_1/imports/Lab3/algorithm_tb_behav.wcfg}
```

2.2.5 Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
synth_1	constrs_1	synth_design Complete!								1021	2490	0.00	0	0
impl_1	constrs_1	route_design Complete, Failed Timing!	0.423	0.000	-0.082	-0.325	0.000	0.159	0	994	2408	0.00	0	0

The best period where the constraints are met is at 12ns with a WNS of 0.423ns. The fastest frequency at which this design can run, taking into account the WNS value, is 86.38MHz

2.2.6 Post-Route Timing Report: Max Delay Path

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.423ns  (required time - arrival time)
  Source:                 INTB_reg[3]/C
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@6.000ns period=12.000ns})
  Destination:            INT2_pipe3_reg[29]/D
                            (rising edge-triggered cell FDRE clocked by clk  {rise@0.000ns fall@6.000ns period=12.000ns})
  Path Group:             clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            12.000ns  (clk rise@12.000ns - clk rise@0.000ns)
  Data Path Delay:        11.549ns  (logic 4.047ns (35.042%)  route 7.502ns (64.958%))
```

From this report, we can identify that the new critical path lies between INTB_reg and INT2_pipe3_reg. This is located in between the input register stage and the third pipeline stage, and the mathematical operation that is within this critical path is the multiplication between inputs B and C.

2.2.7 Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
synth_1	constrs_1	synth_design Complete!								685	2376	0.00	0	2
impl_1	constrs_1	route_design Complete, Failed Timing!	0.739	0.000	-0.058	-0.132	0.000	0.231	0	668	2294	0.00	0	2

The best period where the constraints are met is at 5ns with a WNS of 0.739ns. The fastest frequency at which this design can run, considering the WNS value, is 234.69MHz.

2.2.8 Post-Route Timing Report: Max Delay Path

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.739ns  (required time - arrival time)
  Source:                 my_divider/blk00000003/blk00000710/C
                            (rising edge-triggered cell FDSE clocked by clk  {rise@0.000ns fall@2.500ns period=5.000ns})
  Destination:            my_divider/blk00000003/blk000006ff/D
                            (rising edge-triggered cell FDSE clocked by clk  {rise@0.000ns fall@2.500ns period=5.000ns})
  Path Group:             clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            5.000ns  (clk rise@5.000ns - clk rise@0.000ns)
  Data Path Delay:        4.265ns  (logic 1.787ns (41.901%)  route 2.478ns (58.098%))
```

From this report, I think I can derive that the new critical path lies within the divider IP block component. This is between the first and second pipeline stages, and parallel to the pipeline array. It isn't much of a surprise to see a divider operation becoming part of the critical path, as the division operation is quite demanding in terms of hardware.

2.2.9 Code and Component Statistics

VHDL Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

-- Entity description:
-- The entity implements, with no optimization, a sequence of operations:
--         O <= (A*3 + B*C)/D + C + 5
-- where A,B,C, and D are UNSIGNED vectors of parameterizable size

-- Note 1: There is no particular "meaning" to the equation - it is designed for
--  experimentation with logic optimization for performance
-- Note 2: There is no provision for overflow. Some input vectors can cause
--  overflow and the result will be incorrect.
-- Note 3: Inputs and outputs are registered (rising edge, synchronous reset).
--  This introduces a latency of 2 clock cycles between inputs and outputs.
-- Note 4: D is the divisor in one of the operations, so can never have value 0

entity algorithm is
    generic (data_size : integer := 16); -- defines the size of the data
    Port ( clk : in  STD_LOGIC;
           rst : in  STD_LOGIC;
           -- The four (parameterizable) data inputs
           A : in  STD_LOGIC_VECTOR (data_size-1 downto 0);
           B : in  STD_LOGIC_VECTOR (data_size-1 downto 0);
           C : in  STD_LOGIC_VECTOR (data_size-1 downto 0);
           D : in  STD_LOGIC_VECTOR (data_size-1 downto 0);
           -- Output = (A*3 + B*C)/D + C +5
           O : out  STD_LOGIC_VECTOR (data_size*2-1 downto 0)
           );
end algorithm;

architecture Behavioral of algorithm is

-- Registered inputs
signal INTA, INTB, INTC, INTD : UNSIGNED (data_size-1 downto 0);
-- Internal signals for intermediate operations (note the sizes)
signal INT1 : UNSIGNED (data_size+1 downto 0);   -- INTA + 3
signal INT2 : UNSIGNED (data_size*2-1 downto 0); -- INTB * INTC
signal INT3 : UNSIGNED (data_size*2-1 downto 0); -- INT1 + INT2
signal INT4 : UNSIGNED (data_size*2-1 downto 0); -- INT3 / INTD
-- INT5 needs to be resized to match INTC+1, as this vector size
-- is larger than the required vector size for a decimal '5'
signal INT5 : UNSIGNED (data_size downto 0);   -- [UPDATED] INTC + 5
-- INTO needs to be resized to match INT4, as this is the larger
-- vector size compared to INT5 as we updated that to be smaller
signal INTO : UNSIGNED (data_size*2-1 downto 0); -- [UPDATED] INT4 + INT5

-- We need to define internal signals for after the pipeline stage,
-- these signals will be physically identical to the original ones
-- before the pipeline.
signal INT3_pipe1 : UNSIGNED (data_size*2-1 downto 0);  -- Pipeline 1 output
signal INTD_pipe1 : UNSIGNED (data_size-1 downto 0);    -- Pipeline 1 output
signal INT5_pipe1 : UNSIGNED (data_size downto 0);      -- Pipeline 1 output
signal INT4_pipe2 : UNSIGNED (data_size*2-1 downto 0);  -- Pipeline 2 output
signal INT5_pipe2 : UNSIGNED (data_size downto 0);      -- Pipeline 2 output
signal INT1_pipe3 : UNSIGNED (data_size+1 downto 0);    -- Pipeline 3 output
signal INT2_pipe3 : UNSIGNED (data_size*2-1 downto 0);  -- Pipeline 3 output
signal INT5_pipe3 : UNSIGNED (data_size downto 0);      -- Pipeline 3 output
```

```
signal INTD_pipe3 : UNSIGNED (data_size-1 downto 0);    -- Pipeline 3 output

-- We need an array of INT5 signals for the for-generate pipeline flip flop
-- array.
type INT5_pipeArr is array (0 to 34) of UNSIGNED(data_size downto 0);
signal INT5_pipeArr_int : INT5_pipeArr;
-- Store the output of the pipeline array to this output signal.
signal INT5_pipeArr_out : UNSIGNED (data_size downto 0);

-- Divider IP component instantiation
component divider
port (
    clk : in STD_LOGIC;
    sclr : in STD_LOGIC;
    rfd : out STD_LOGIC;
    dividend : in STD_LOGIC_VECTOR(31 downto 0);
    divisor : in STD_LOGIC_VECTOR(15 downto 0);
    quotient : out STD_LOGIC_VECTOR(31 downto 0);
    fractional : out STD_LOGIC_VECTOR(15 downto 0)
);
end component;

begin
-- Port map the IP divider block
my_divider : divider
port map (
    clk => clk,
    sclr => rst,
    dividend => STD_LOGIC_VECTOR(INT3_pipe1),
    divisor => STD_LOGIC_VECTOR(INTD_pipe1),
    unsigned(quotient) => INT4
);

-- Input registers (D-type, rising edge, synchronous reset)
input_regs: process (clk) is
begin
  if rising_edge(clk) then
    if rst = '1' then
      INTA <= (others => '0');
      INTB <= (others => '0');
      INTC <= (others => '0');
      INTD <= to_unsigned(1,INTD'length); -- type conversion notation
    else
      INTA <= unsigned(A);
      INTB <= unsigned(B);
      INTC <= unsigned(C);
      INTD <= unsigned(D);
    end if;
  end if;
end process input_regs;
```

```
-- Pipeline 1 resgisters (D-type, rising edge, synchronous reset)
pipe1_regs: process (clk) is
begin
  if rising_edge(clk) then
    if rst = '1' then
      INT3_pipe1 <= (others => '0');
      INTD_pipe1 <= to_unsigned(1,INTD_pipe3'length); -- type conversion notation
      INT5_pipe1 <= (others => '0');
    else
      INT3_pipe1 <= INT3;
      INTD_pipe1 <= INTD_pipe3;
      INT5_pipe1 <= INT5_pipe3;
    end if;
  end if;
end process pipe1_regs;


-- Pipeline 2 resgisters (D-type, rising edge, synchronous reset)
pipe2_regs: process (clk) is
begin
  if rising_edge(clk) then
    if rst = '1' then
      INT4_pipe2 <= (others => '0');
      INT5_pipe2 <= (others => '0');
    else
      INT4_pipe2 <= INT4;
      INT5_pipe2 <= INT5_pipeArr_out;
    end if;
  end if;
end process pipe2_regs;

-- Pipeline 3 resgisters (D-type, rising edge, synchronous reset)
pipe3_regs: process (clk) is
begin
  if rising_edge(clk) then
    if rst = '1' then
      INT1_pipe3 <= (others => '0');
      INT2_pipe3 <= (others => '0');
      INT5_pipe3 <= (others => '0');
      INTD_pipe3 <= (others => '0');
    else
      INT1_pipe3 <= INT1;
      INT2_pipe3 <= INT2;
      INT5_pipe3 <= INT5;
      INTD_pipe3 <= INTD;
    end if;
  end if;
end process pipe3_regs;
```

```
-- For-Generate Loop to add a delay for INT5_pipe1 to match the
-- latency of the divider with an array of 34 pipelines. First
-- we need to link INT5_pipe1 to the first signal bus in
-- INT5_pipe1_array_int, then the last signal of INT5_pipe1_array_int
-- to the output INT5_pipeArr.
INT5_pipeArr_int(0) <= INT5_pipe1;
pipeline_array : for i in 1 to 34 generate
    pipeline: process (clk) is
    begin
      if rising_edge(clk) then
        if rst = '1' then
          INT5_pipeArr_int(i) <= (others => '0');
        else
          INT5_pipeArr_int(i) <= INT5_pipeArr_int(i-1);
        end if;
      end if;
    end process pipeline;
end generate;
INT5_pipeArr_out <= INT5_pipeArr_int(34);   -- Map the output to signal

-- Mathematical operations on the data (combinational)
INT1 <= INTA * to_unsigned(3, 2);
INT2 <= INTB * INTC;
INT3 <= INT1_pipe3 + INT2_pipe3;
-- This logic has been moved forward along side the multiplication
-- so that the adders aren't exactly one after the other operations
-- in INT1 and INT2. In the previous design, this logic would've
-- been carried out by INT0, we need to make sure that the to_unsigned
-- function has the correct size parameter (width of INTC + 1)
INT5 <= INTC + to_unsigned(5, INTC'length+1);
-- Bringing the INT5 logic forward needs to be implemented, therefore
-- INT0 has been adapted to take into account this change. Critical
-- path is reduced as the INT5 adder is along side the rest of the
-- logic, INT0 doesn't have to wait for INT5 like before in the
-- pre-modifications design.
INT0 <= INT4_pipe2 + INT5_pipe2;

-- Input registers (D-type, rising edge, synchronous reset)
output_regs: process (clk) is
begin
  if rising_edge(clk) then
    if rst = '1' then
      O <= (others => '0');
    else
      O <= std_logic_vector(INT0);
    end if;
  end if;
end process output_regs;


end Behavioral;
```

RTL Component Statistics

```
Detailed RTL Component Info :
+---Adders :
           2 Input     32 Bit       Adders := 1
           2 Input     17 Bit       Adders := 1
+---Registers :
                       32 Bit    Registers := 2
                       17 Bit    Registers := 37
                       16 Bit    Registers := 6
```

RTL Hierarchical Component Statistics

```
Hierarchical RTL Component report
Module algorithm
Detailed RTL Component Info :
+---Adders :
           2 Input     32 Bit       Adders := 1
           2 Input     17 Bit       Adders := 1
+---Registers :
                       32 Bit    Registers := 2
                       17 Bit    Registers := 37
                       16 Bit    Registers := 6
```