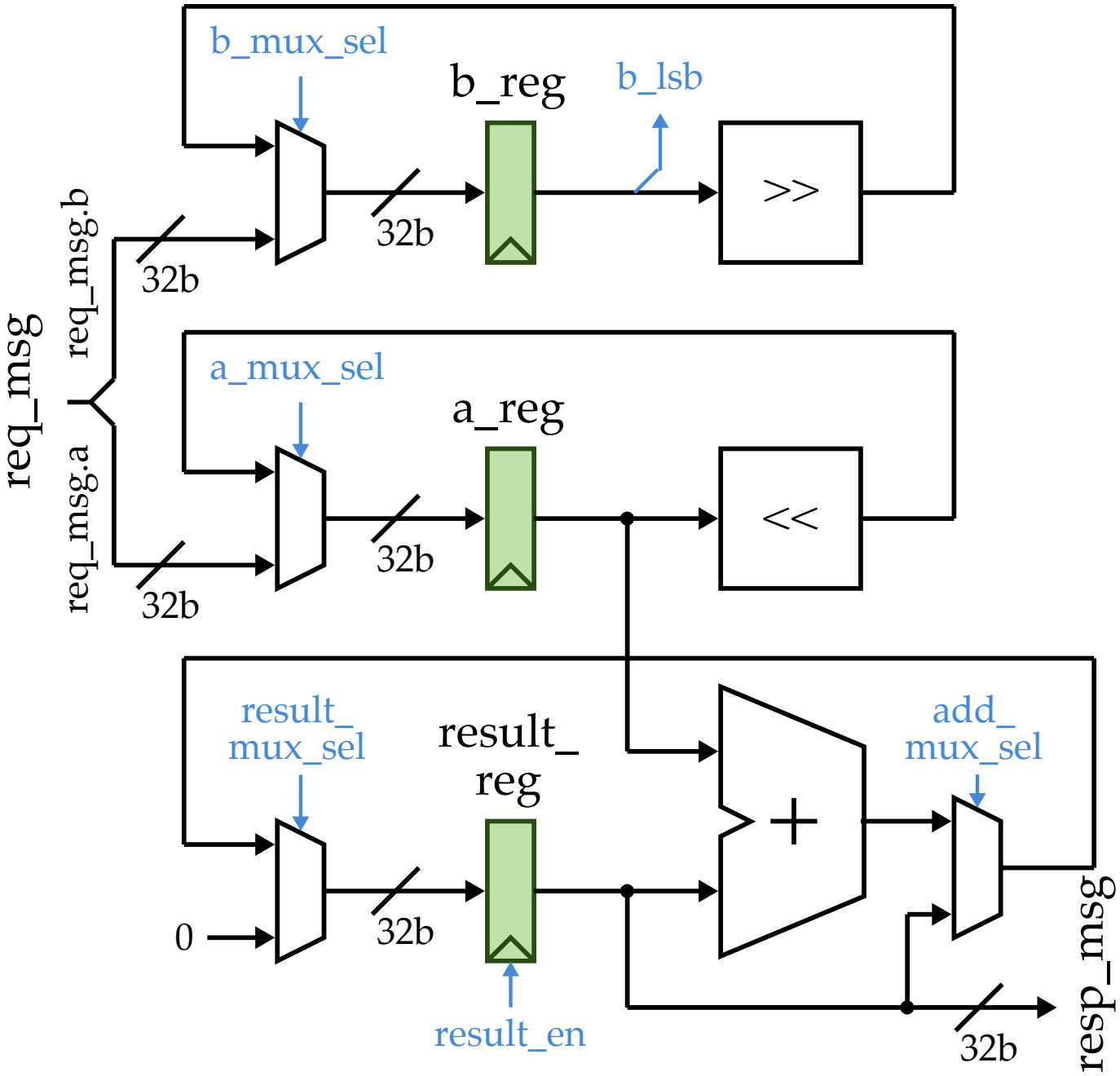
Path 1: MET Setup Check with Pin dpath/result_reg/out_reg_26_/CK
Endpoint: dpath/result_reg/out_reg_26_/D (^) checked with leading edge of 'ideal_clock'
Beginpoint: dpath/a_reg/out_reg_8_/Q   (^) triggered by leading edge of 'ideal_clock'
Path Groups: {Reg2Reg}
Analysis View: analysis_default
Other End Arrival Time   -0.027
- Setup   0.030
+ Phase Shift   0.500
+ CPPR Adjustment   0.008
= Required Time   0.452
- Arrival Time   0.443
= Slack Time   0.009
     Clock Rise Edge   0.000
     + Drive Adjustment   0.020
     + Source Insertion Delay   -0.072
     = Beginpoint Arrival Time   -0.052
     Timing Path:

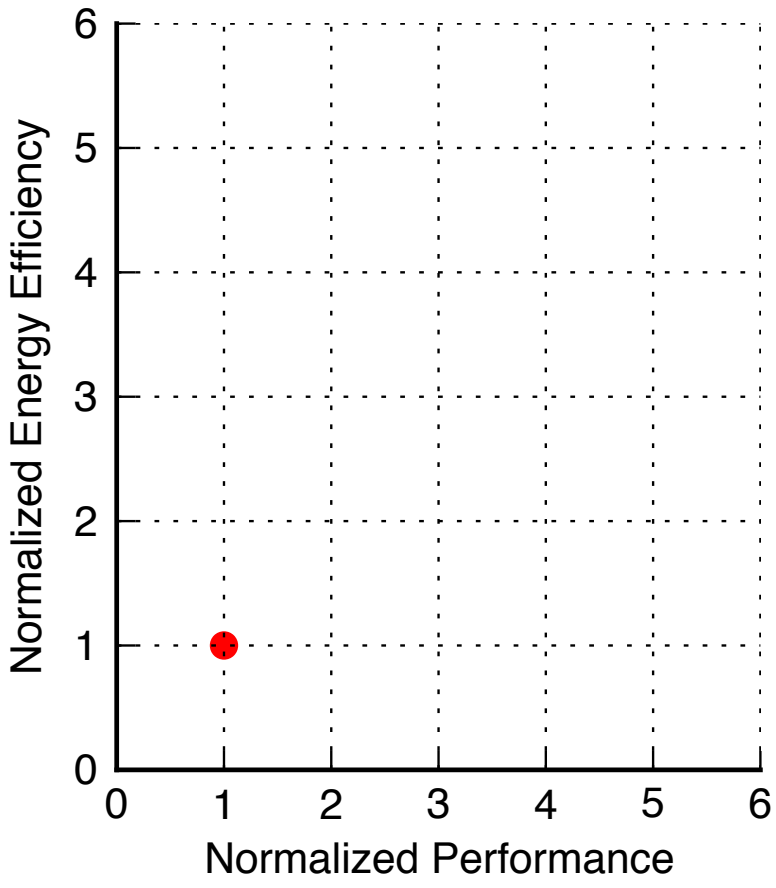
Instance	Arc	Cell	Slew	Delay	Arrival Time	Required Time
	clk[0] ^		0.020		-0.052	-0.043
dpath/CTS_ccl_a_BUF_ideal_clock_G0_L1_1		CLKBUF_X3	0.020	0.001	-0.051	-0.043
dpath/CTS_ccl_a_BUF_ideal_clock_G0_L1_1	A ^ -> Z ^	CLKBUF_X3	0.039	0.071	0.020	0.029
dpath/a_reg/out_reg_8_		DFF_X1	0.039	0.001	0.021	0.030
dpath/a_reg/out_reg_8_	CK ^ -> Q ^	DFF_X1	0.023	0.115	0.136	0.145
dpath/add/DP_OP_5J1_122_6202/U404_dup		NAND2_X1	0.023	0.000	0.136	0.145
dpath/add/DP_OP_5J1_122_6202/U404_dup	A2 ^ -> ZN v	NAND2_X1	0.016	0.021	0.158	0.166
dpath/add/DP_OP_5J1_122_6202/FE_RC_4_0		OAI21_X2	0.016	0.000	0.158	0.166
dpath/add/DP_OP_5J1_122_6202/FE_RC_4_0	B2 v -> ZN ^	OAI21_X2	0.028	0.045	0.203	0.212
dpath/add/DP_OP_5J1_122_6202/FE_RC_6_0		AOI21_X2	0.028	0.000	0.203	0.212
dpath/add/DP_OP_5J1_122_6202/FE_RC_6_0	B1 ^ -> ZN v	AOI21_X2	0.013	0.022	0.225	0.234
dpath/add/DP_OP_5J1_122_6202/U477		OAI21_X1	0.013	0.000	0.225	0.234
dpath/add/DP_OP_5J1_122_6202/U477	B1 v -> ZN ^	OAI21_X1	0.020	0.030	0.255	0.264
dpath/add/DP_OP_5J1_122_6202/FE_RC_3_0		INV_X1	0.020	0.000	0.256	0.264
dpath/add/DP_OP_5J1_122_6202/FE_RC_3_0	A ^ -> ZN v	INV_X1	0.008	0.013	0.268	0.277
dpath/add/DP_OP_5J1_122_6202/FE_RC_0_0		NAND2_X2	0.008	0.000	0.268	0.277
dpath/add/DP_OP_5J1_122_6202/FE_RC_0_0	A2 v -> ZN ^	NAND2_X2	0.013	0.020	0.288	0.297
dpath/add/DP_OP_5J1_122_6202/FE_RC_1_0		INV_X4	0.013	0.000	0.288	0.297
dpath/add/DP_OP_5J1_122_6202/FE_RC_1_0	A ^ -> ZN v	INV_X4	0.010	0.017	0.305	0.314
dpath/add/DP_OP_5J1_122_6202/U599		OAI21_X1	0.011	0.002	0.308	0.317
dpath/add/DP_OP_5J1_122_6202/U599	B1 v -> ZN ^	OAI21_X1	0.027	0.037	0.344	0.353
dpath/add/DP_OP_5J1_122_6202/FE_RC_4_1		NAND2_X1	0.027	0.000	0.344	0.353
dpath/add/DP_OP_5J1_122_6202/FE_RC_4_1	A1 ^ -> ZN v	NAND2_X1	0.010	0.017	0.361	0.370
dpath/add/DP_OP_5J1_122_6202/FE_RC_3_1		OAI21_X1	0.010	0.000	0.361	0.370
dpath/add/DP_OP_5J1_122_6202/FE_RC_3_1	A v -> ZN ^	OAI21_X1	0.022	0.023	0.384	0.392
dpath/add_mux/U83		NAND2_X1	0.022	0.000	0.384	0.393
dpath/add_mux/U83	A1 ^ -> ZN v	NAND2_X1	0.012	0.016	0.400	0.409
dpath/add_mux/U85		NAND2_X1	0.012	0.000	0.400	0.409
dpath/add_mux/U85	A1 v -> ZN ^	NAND2_X1	0.009	0.014	0.414	0.423
dpath/result_mux/U11		AND2_X1	0.009	0.000	0.414	0.423
dpath/result_mux/U11	A1 ^ -> ZN ^	AND2_X1	0.009	0.029	0.443	0.452
dpath/result_reg/out_reg_26_		DFF_X1	0.009	0.000	0.443	0.452

     Clock Rise Edge   0.000
     + Drive Adjustment   0.012
     + Source Insertion Delay   -0.072
     = Beginpoint Arrival Time   -0.061
     Other End Path:

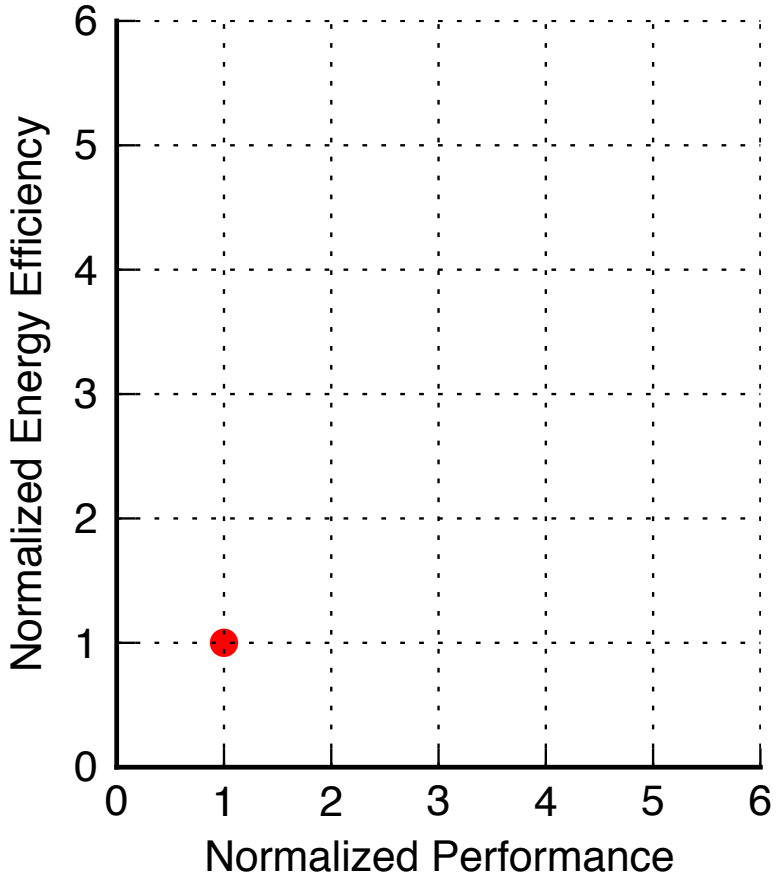
Instance	Arc	Cell	Slew	Delay	Arrival Time	Required Time
	clk[0] ^		0.011		-0.061	-0.069
dpath/result_reg/clk_gate_out_reg/latch		CLKGATETST_X4	0.011	0.001	-0.059	-0.068
dpath/result_reg/clk_gate_out_reg/latch	CK ^ -> GCK ^	CLKGATETST_X4	0.016	0.031	-0.028	-0.037
dpath/result_reg/out_reg_26_		DFF_X1	0.016	0.001	-0.027	-0.035


b_mux_sel
b_reg
b_lsb
req_msg
req_msg.b
32b
32b
>>
a_mux_sel
a_reg
req_msg.a
32b
32b
<<
result_
mux_sel
result_
reg
add_
mux_sel
0
32b
result_en
32b
resp_msg

```
vsrc       = IntMulFixedLatPRTL_0x791afe0d4d8c.v
input      = imul-rtl-fixed-small
area       = 1168.272 # um^2
constraint = 1.0 # ns
slack      = 0.009 # ns
exec_time  = 1705 # cycles
power      = 3.958 # mW
energy     = 6.8158739 # nJ

vsrc       = IntMulFixedLatPRTL_0x791afe0d4d8c.v
input      = imul-rtl-fixed-large
area       = 9178 # um^2
constraint = 1.0 # ns
exec_time  = 1705 # cycles
power      = 6.532 # mW
energy     = 11.2484306 # nJ

vsrc       = IntMulVarLatPRTL_0x791afe0d4d8c.v
input      = imul-rtl-var-small
area       = 13300 # um^2
constraint = 1.0 # ns
exec_time  = 398 # cycles
power      = 5.231 # mW
energy     = 2.10275738 # nJ

vsrc       = IntMulVarLatPRTL_0x791afe0d4d8c.v
input      = imul-rtl-var-large
area       = 13300 # um^2
constraint = 1.0 # ns
exec_time  = 1371 # cycles
power      = 7.536 # mW
energy     = 10.43517456 # nJ
```



Energy Efficiency vs. Performance for Small Random Inputs



Energy Efficiency vs. Performance for Large Random Inputs

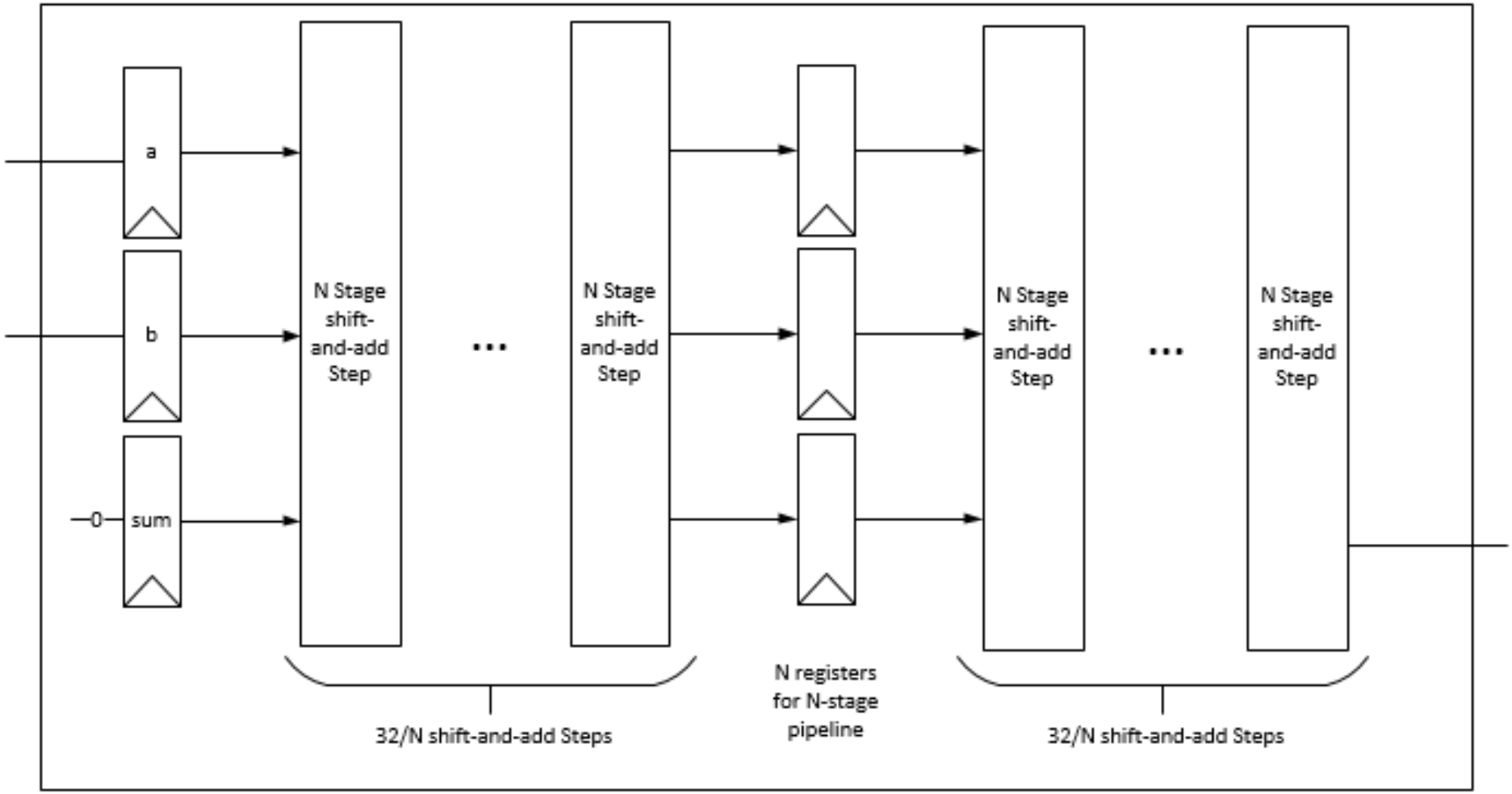


Figure 3.1 - N-stage Pipelined Multiplier: This multiplier consists of a variable number of stages based on an input parameter. Each stage is bookended by registers, and a stage consists of 32/N shift-and-add steps (see **Figure 3.2** below). The pipelining allows multiple multiplication operations to being partially computed each cycle in different stages of the multiplier to allow a throughput of 1 multiplication operation per cycle ideally.

```
../lab1-imul/IntMulNstageInelasticRTL_test.py::test_8stage ()
  2:                    >                    [........]          >
  3:                    >                    [........]          >
  4: 00000001:0000002f > 00000001:0000002f [........]          >
  5:                    >                    [1.......]          >
  6: 0000003c:0000002b > 0000003c:0000002b [.1......]          >
  7:                    >                    [1.1.....]          >
  8: 0000003d:00000049 > 0000003d:00000049 [.1.1....]          >
  9:                    >                    [1.1.1...]          >
 10:                    >                    [.1.1.1..]          >
 11: 0000003f:00000061 > 0000003f:00000061 [..1.1.1.]          >
 12:                    >                    [1..1.1.1] 0000002f > 0000002f
 13: #                  > #                  [.1..1.1.] .        > .
 14: #                  > #                  [.1..1.1.] .        > .
 15: #                  > #                  [.1..1.1.] .        > .
 16: #                  > #                  [.1..1.1.] .        > .
 17: #                  > #                  [.1..1.1.] .        > .
 18: #                  > #                  [.1..1.1.] .        > .
 19: 00000021:00000024 > 00000021:00000024 [.1..1.1.]          >
 20:                    >                    [1.1..1.1] 00000a14 > 00000a14
 21: .                  > .                  [.1.1..1.] .        > .
 22: .                  > .                  [.1.1..1.] .        > .
 23: #                  > #                  [.1.1..1.] .        > .
 24: #                  > #                  [.1.1..1.] .        > .
 25: #                  > #                  [.1.1..1.] .        > .
 26: #                  > #                  [.1.1..1.] .        > .
 27: #                  > #                  [.1.1..1.] .        > .
 28: #                  > #                  [.1.1..1.] .        > .
 29: #                  > #                  [.1.1..1.] .        > .
 30: 00000043:00000049 > 00000043:00000049 [.1.1..1.]          >
 31:                    >                    [1.1.1..1] 00001165 > 00001165
 32: .                  > .                  [.1.1.1..] .        > .
 33: .                  > .                  [.1.1.1..] .        > .
 34: #                  > #                  [.1.1.1..] .        > .
 35: #                  > #                  [.1.1.1..] .        > .
 36: #                  > #                  [.1.1.1..] .        > .
 37: #                  > #                  [.1.1.1..] .        > .
 38: 00000052:00000026 > 00000052:00000026 [.1.1.1..]          >
 39:                    >                    [1.1.1.1.]          >
 40:                    >                    [.1.1.1.1] 000017df > 000017df
 41: .                  > .                  [..1.1.1.] .        > .
 42: 00000033:0000001b > 00000033:0000001b [..1.1.1.]          >
 43:                    >                    [1..1.1.1] 000004a4 > 000004a4
 44: #                  > #                  [.1..1.1.] .        > .
```

Figure 4.4 - Trace for Inelastic 8-stage Pipelined Multiplier: The eight dots/ones in the square brackets specify in the value in the stage is invalid/valid. Notice that when the output is not ready (denoted with a dot on the right-most column), the pipeline does not have any valid computations progressing anywhere in the pipeline.

```
../lab1-imul/IntMulNstageRTL_test.py::test_8stage ()
  2:                    >                    [........]          >
  3:                    >                    [........]          >
  4: 00000001:0000002f > 00000001:0000002f [........]          >
  5:                    >                    [1.......]          >
  6: 0000003c:0000002b > 0000003c:0000002b [.1......]          >
  7:                    >                    [1.1.....]          >
  8: 0000003d:00000049 > 0000003d:00000049 [.1.1....]          >
  9:                    >                    [1.1.1...]          >
 10:                    >                    [.1.1.1..]          >
 11: 0000003f:00000061 > 0000003f:00000061 [..1.1.1.]          >
 12:                    >                    [1..1.1.1] 0000002f > 0000002f
 13: 00000021:00000024 > 00000021:00000024 [.1..1.1.] .        > .
 14:                    >                    [1.1..1.1] #        > #
 15:                    >                    [.1.1..11] #        > #
 16: 00000043:00000049 > 00000043:00000049 [..1.1.11] #        > #
 17:                    >                    [1..1.111] #        > #
 18:                    >                    [.1..1111] #        > #
 19:                    >                    [..1.1111] 00000a14 > 00000a14
 20: 00000052:00000026 > 00000052:00000026 [...1.111] #        > #
 21:                    >                    [1...1111] #        > #
 22:                    >                    [.1..1111] #        > #
 23:                    >                    [..1.1111] #        > #
 24: 00000033:0000001b > 00000033:0000001b [...11111] #        > #
 25:                    >                    [1..11111] #        > #
 26:                    >                    [.1.11111] 00001165 > 00001165
 27:                    >                    [..1.1111] #        > #
 28: 00000064:00000062 > 00000064:00000062 [...11111] 000017df > 000017df
 29:                    >                    [1...1111] #        > #
 30: 00000011:0000001f > 00000011:0000001f [.1..1111] 000004a4 > 000004a4
 31:                    >                    [1.1..111] #        > #
 32: 0000000d:00000029 > 0000000d:00000029 [.1.1.111] #        > #
 33:                    >                    [1.1.1111] #        > #
 34:                    >                    [.1.11111] #        > #
 35: 00000039:0000003a > 00000039:0000003a [..111111] #        > #
 36:                    >                    [1.111111] #        > #
 37:                    >                    [.1111111] #        > #
 38:                    >                    [.1111111] #        > #
 39: 0000000d:00000064 > 0000000d:00000064 [.1111111] #        > #
 40: .                  > .                  [11111111] #        > #
 41: #                  > #                  [11111111] #        > #
 42: #                  > #                  [11111111] #        > #
 43: #                  > #                  [11111111] #        > #
 44: 00000044:00000000 > 00000044:00000000 [11111111] 0000131b > 0000131b
```

Figure 4.3 - Trace for Elastic 8-stage Pipelined Multiplier: The eight dots/ones between the square brackets specify if the value in the stage is a valid multiplication operation or not. Notice that when the output is not ready (denoted with a # in the far right column) the pipeline squashes bubbles in the pipeline (the dots in the square brackets disappear).