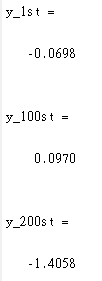
1. Matlab program: Use data broadcast form to implement this filter. Show the output of this filter (confirm your answer with TAs). Show the 1st, 100th and 200th output values in your report.

Result:

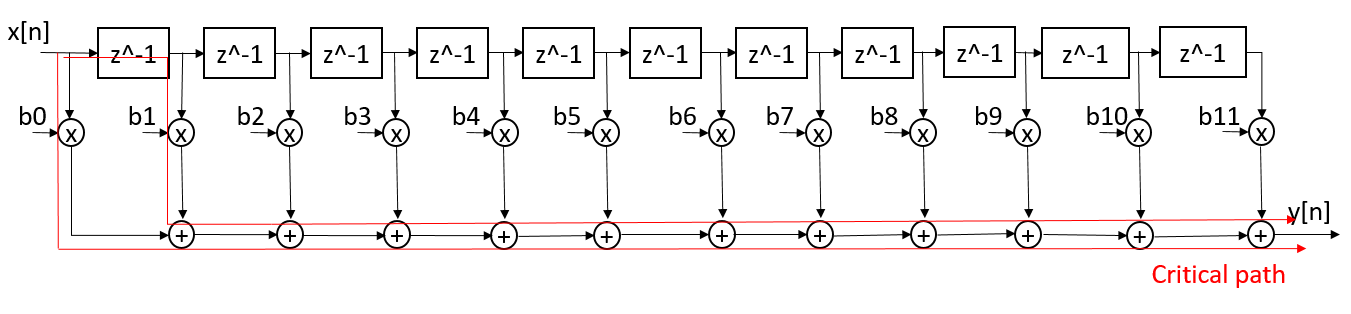


1. Following (1), implement this filter via Verilog. Show the output of this filter (confirm your answer with TAs). Let the clock rate be 100MHz. Synthesize the filter. Then show the gate counts and indicate the critical path.

Waveform of output



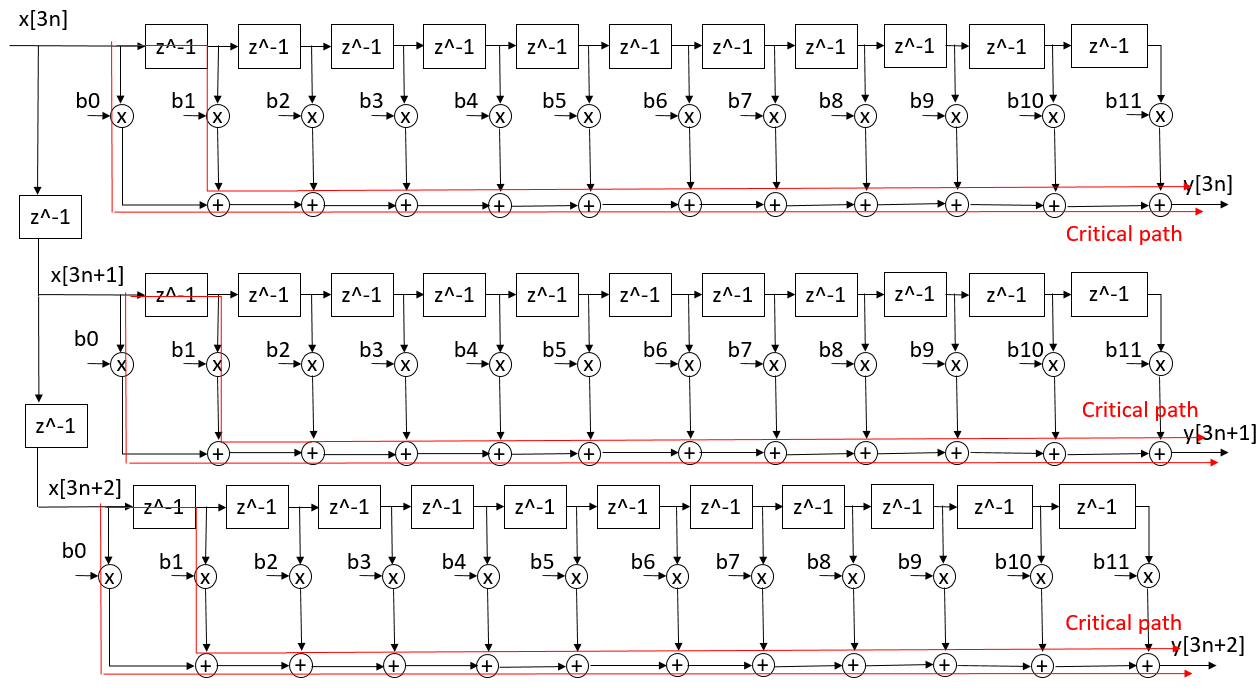
Critical path:



Gate counts: 11 adders & 12 multipliers

1. Let the parallel processing factor be 3 and use the fast algorithm in Ch9. Implement this parallel processed filter using Matlab and Verilog. Show the output of these two implementations. The results of these two implementations should be the same (confirm your answer with TAs).

Architecture



Gate Counts:33 adders & 36 multipliers

相較(2)增加了22個adder與24個multiplier

1. Following (3), use the CSD subexpression elimination to further reduce the gate count of the polyphaser filters.

b0 = 0.00101’001010100001

b1 = 0.000000001010100010

b2 = 0.001’01010101’01’0001

b3 = 0.0001’010001’0100100

b4 = 0.0101’001’00001’00001’

b5 = 0.101’0101001’0001’010

b6 = 0.101’0101001’0001’010

b7 = 0.0101’001’00001’00001’

b8 = 0.0001’010001’0100100

b9 = 0.001’01010101’01’0001

b10 = 0.000000001010100010

b11 = 0.00101’001010100001

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 0 | 1’ | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1’ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1’ | 0 | 1’ | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1’ | 0 | 1 | 0 | 0 | 0 | 1’ | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1’ | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ |
| 0 | 1 | 0 | 1’ | 0 | 1 | 0 | 1 | 0 | 0 | 1’ | 0 | 0 | 0 | 1’ | 0 | 1 | 0 |
| 0 | 1 | 0 | 1’ | 0 | 1 | 0 | 1 | 0 | 0 | 1’ | 0 | 0 | 0 | 1’ | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1’ | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ |
| 0 | 0 | 0 | 0 | 1’ | 0 | 1 | 0 | 0 | 0 | 1’ | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1’ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1’ | 0 | 1’ | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1’ | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

X2 = x1 - x1 >> 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | -2 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | -2 | 0 | 0 | 0 | 0 | 0 | -2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ |
| 0 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1’ | 0 | 0 | 0 | -2 | 0 | 0 | 0 |
| 0 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1’ | 0 | 0 | 0 | -2 | 0 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ |
| 0 | 0 | 0 | 0 | -2 | 0 | 0 | 0 | 0 | 0 | -2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | -2 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

X3 = x2+x1>>5

X4 = x3+x1>>7

X5 = x4+x1>>9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | -2 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | -2 | 0 | 0 | 0 | 0 | 0 | -2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ |
| 0 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1’ | 0 | 0 | 0 | -2 | 0 | 0 | 0 |
| 0 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1’ | 0 | 0 | 0 | -2 | 0 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 0 | 1’ |
| 0 | 0 | 0 | 0 | -2 | 0 | 0 | 0 | 0 | 0 | -2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | -2 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 1’ | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

X6 = x5+x1>>14

=> y = x6>>2

+x1[-1]>>8 + x1[-1]>>10 + x1[-1]>>12 + x1[-1]>>15

-x2[-2]>>2+x1[-2]>>6+x2[-2]>>8 -x1[-2]>>12 + x1[-2]>>16

-x2[-3]>>3 + x2[-3]>>9 + x1[-3]>>14

+x2[-4]>>1 -x1[-4]>>5 -x1[-4]>>10 -x1[-4]>>15

+x2[-5]+x1[-5]>>4+x1[-5]>>6 -x1[-5]>>9 -x2[-5]>>13

+x2[-6]+x1[-6]>>4+x1[-6]>>6 -x1[-6]>>9 -x2[-6]>>13

+x2[-7]>>1 -x1[-7]>>5 -x1[-7]>>10 -x1[-7]>>15

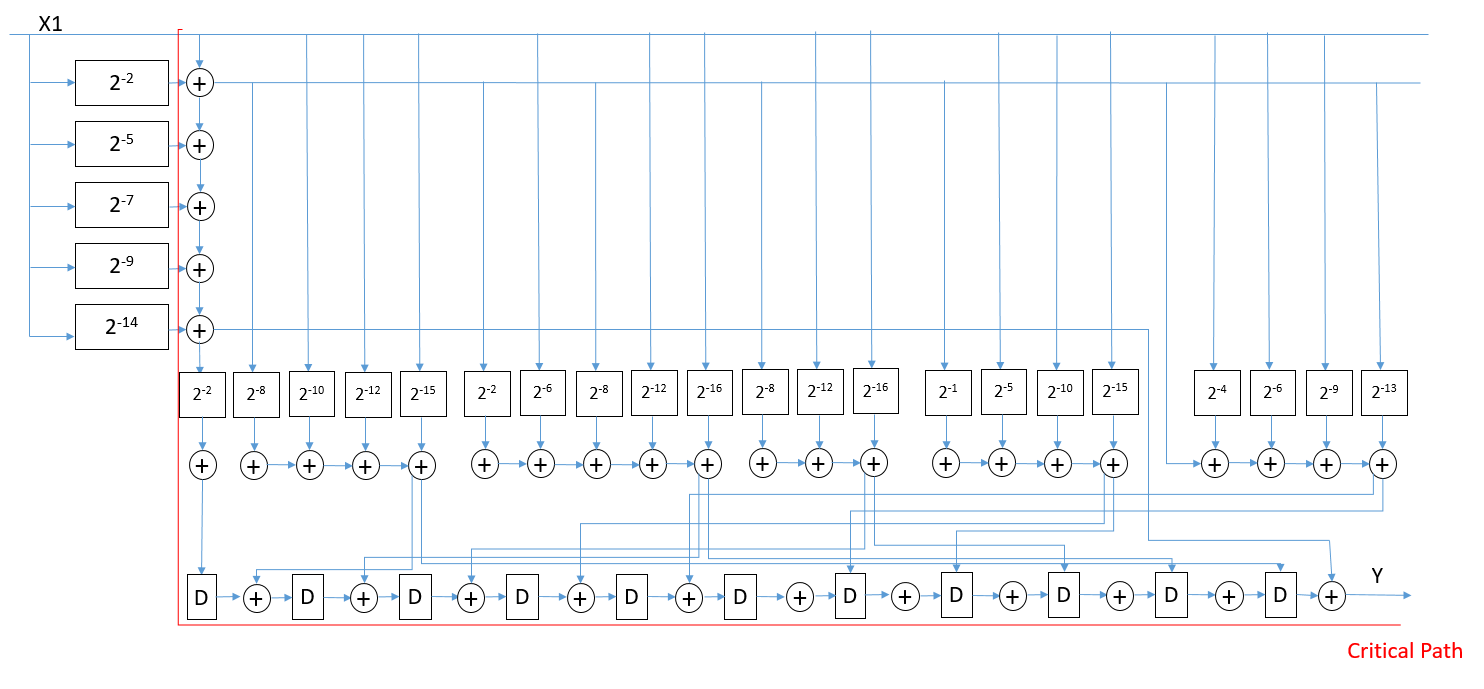
-x2[-8]>>3 + x2[-8]>>9 + x1[-8]>>14

-x2[-9]>>2+x1[-9]>>6+x2[-9]>>8 -x1[-9]>>12 + x1[-9]>>16

+x1[-10]>>8 + x1[-10]>>10 + x1[-10]>>12 + x1[-10]>>15

+x6[-11]>>2

Architecture and critical path:

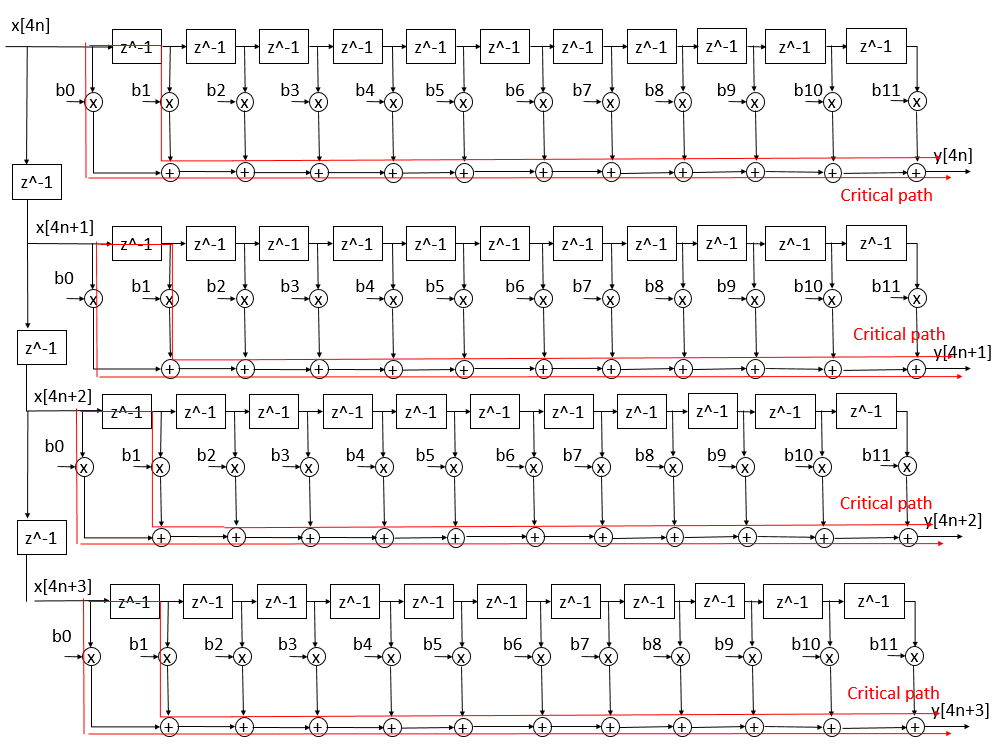


Gate counts: 26 shifts & 37 adders

Originally are 68 shifts & 66 adders

* 42 shifts & 29 adders saved

1. Now let the parallel processing factor be 4. Repeat (3).



Gate Counts:44 adders & 48 multipliers

相較(2)增加了33個adder與36個multiplier