2025 DCS Lab09
Pipeline/FIR filter

吳俊諺

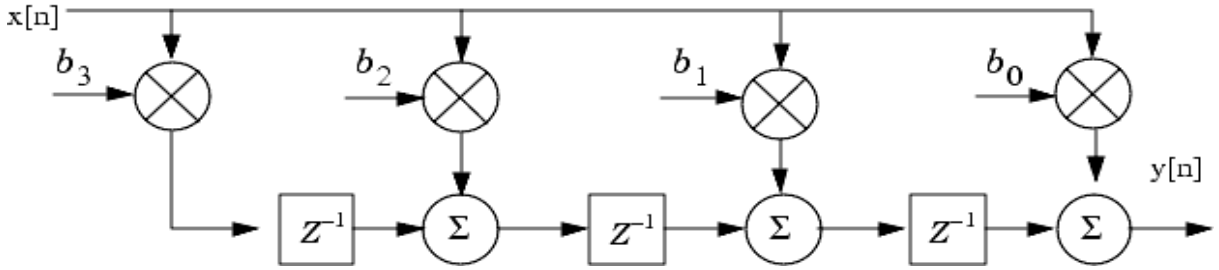
Purpose

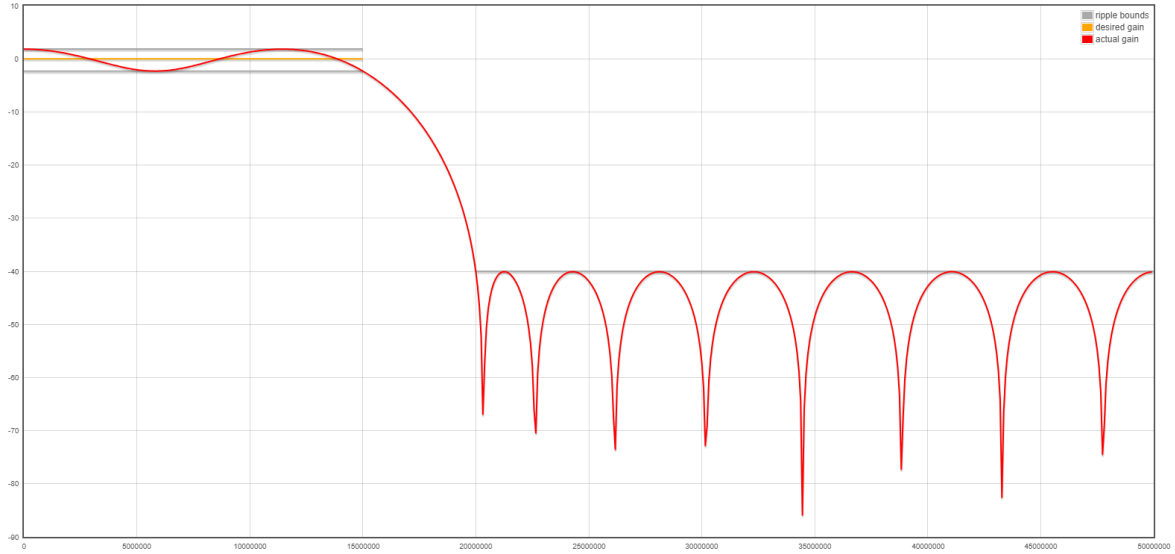
- Learn how to write a pipeline architecture in Verilog.
- FIR filter’s hardware implementation

Low-pass FIR filter

- FIR filter Block Diagram
- 數位低通濾波器
- 相較於類比低通濾波器來說製作難易度低、成本低。
- 硬體上只需要暫存器、乘法器和加法器即可實現濾波效果。

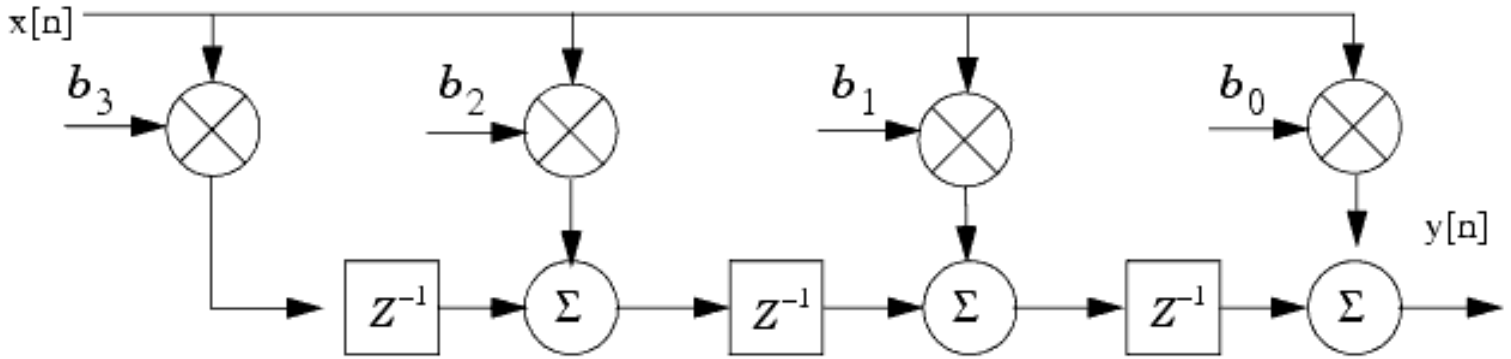
$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3]$$





3rd-Order FIR Filter

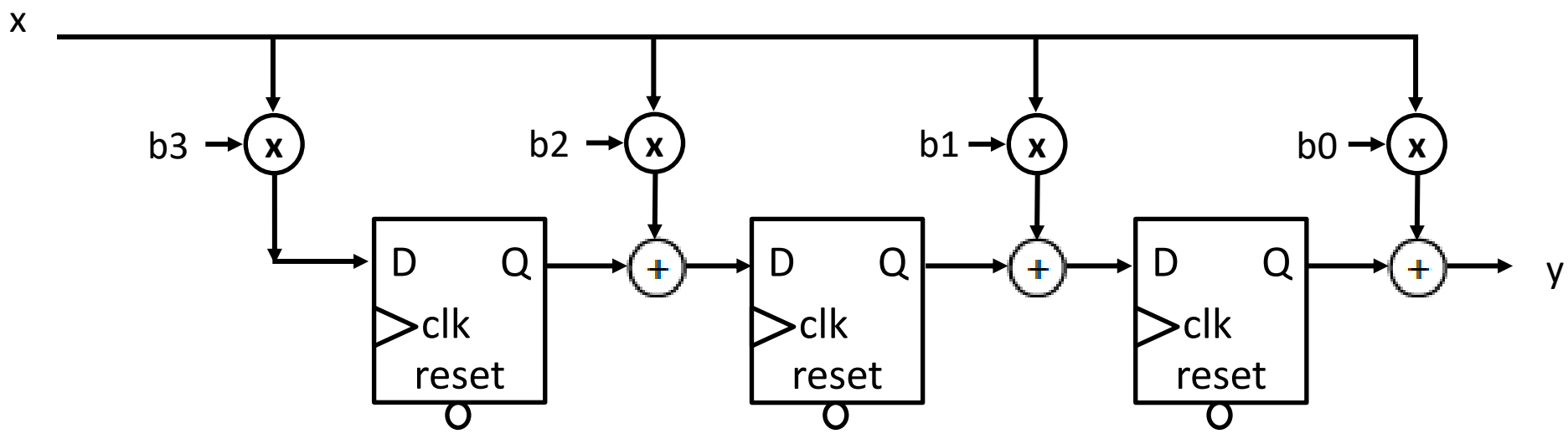
- 這次Lab要實作一個三階FIR Filter： $y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3]$
- 3 ADDs, 4 MULTs如果放在同一個cycle算 => critical path太長
- Solution: **Pipeline!**



Dataflow

- The FIR circuit:

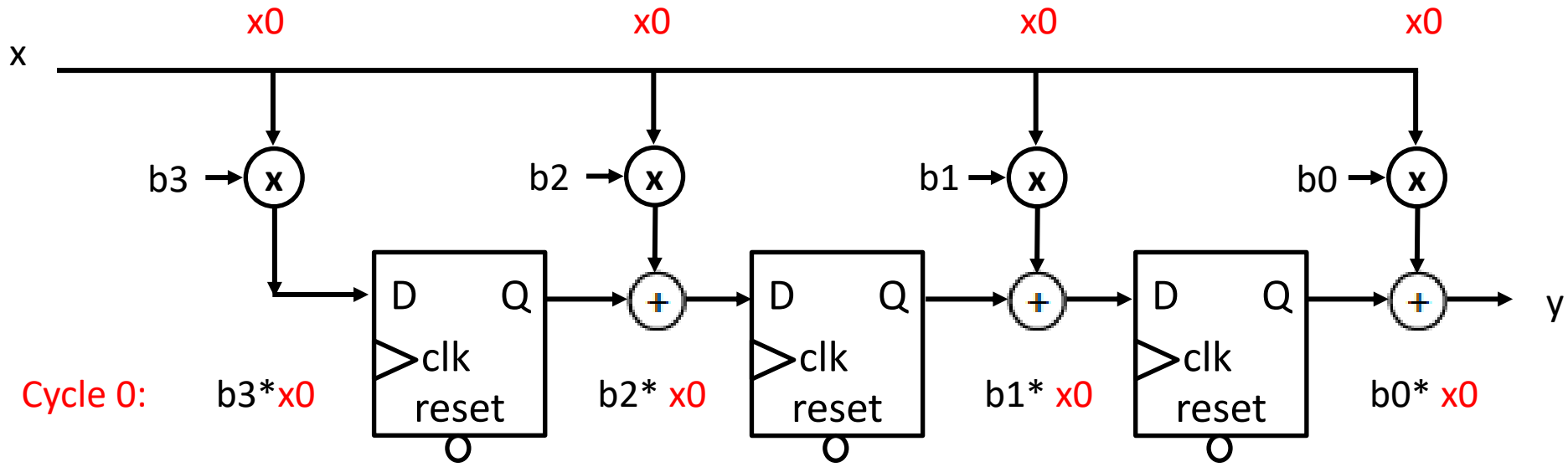
$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3]$$



Dataflow

- Cycle 0:

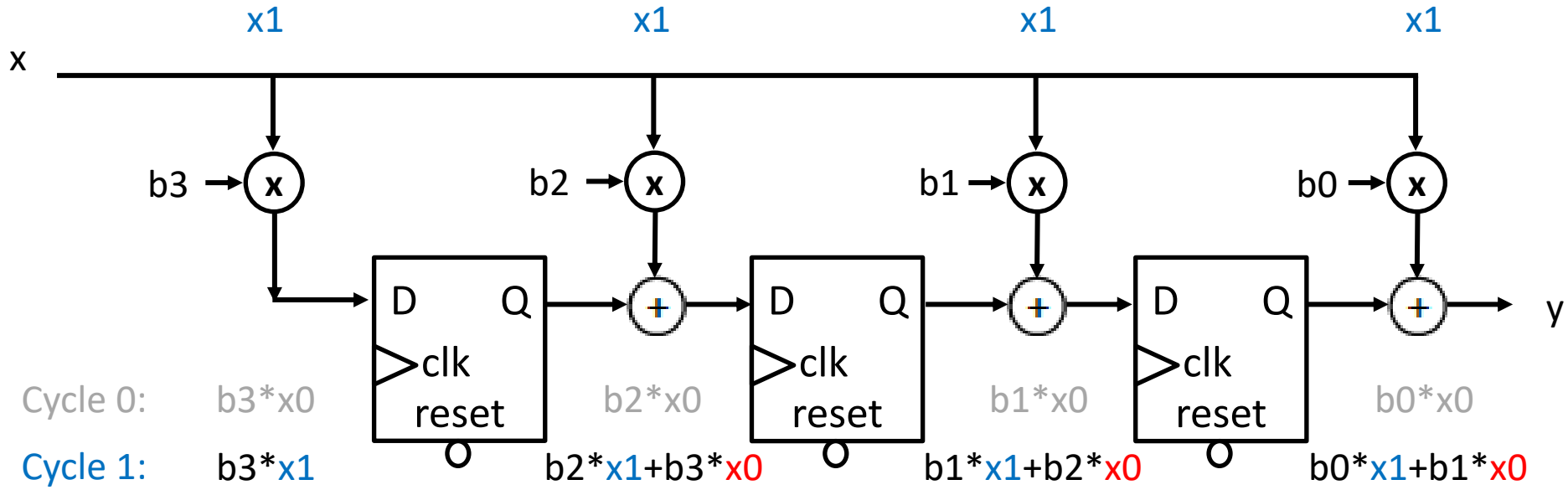
$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3]$$



Dataflow

- Cycle 1:

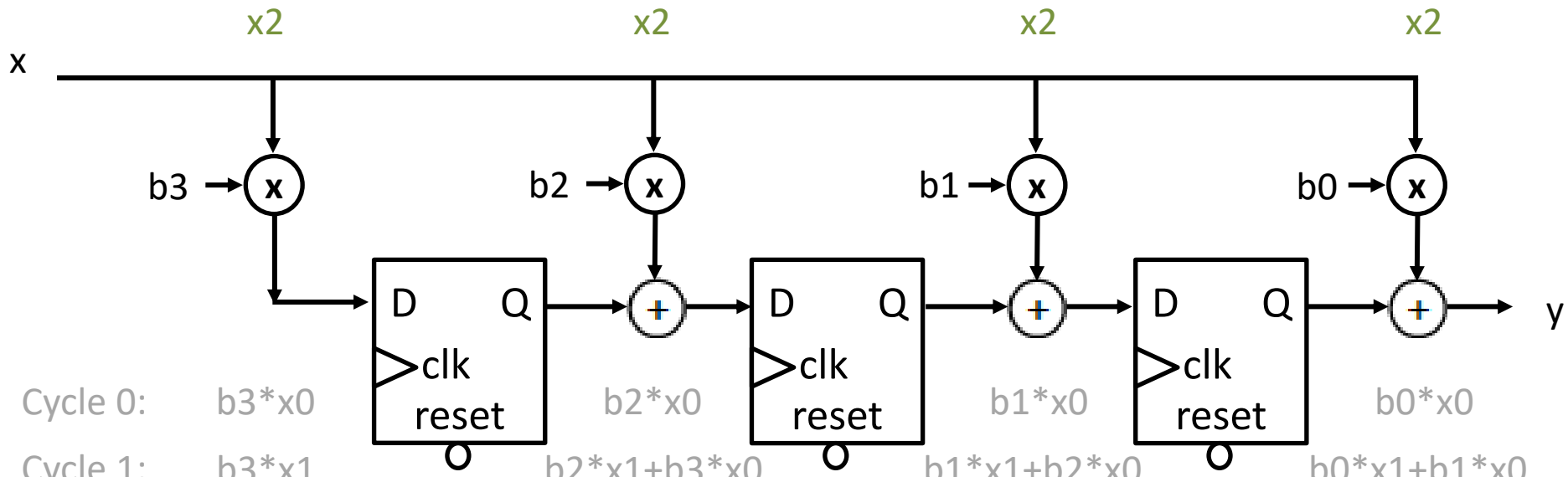
$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3]$$



Dataflow

- Cycle 2:

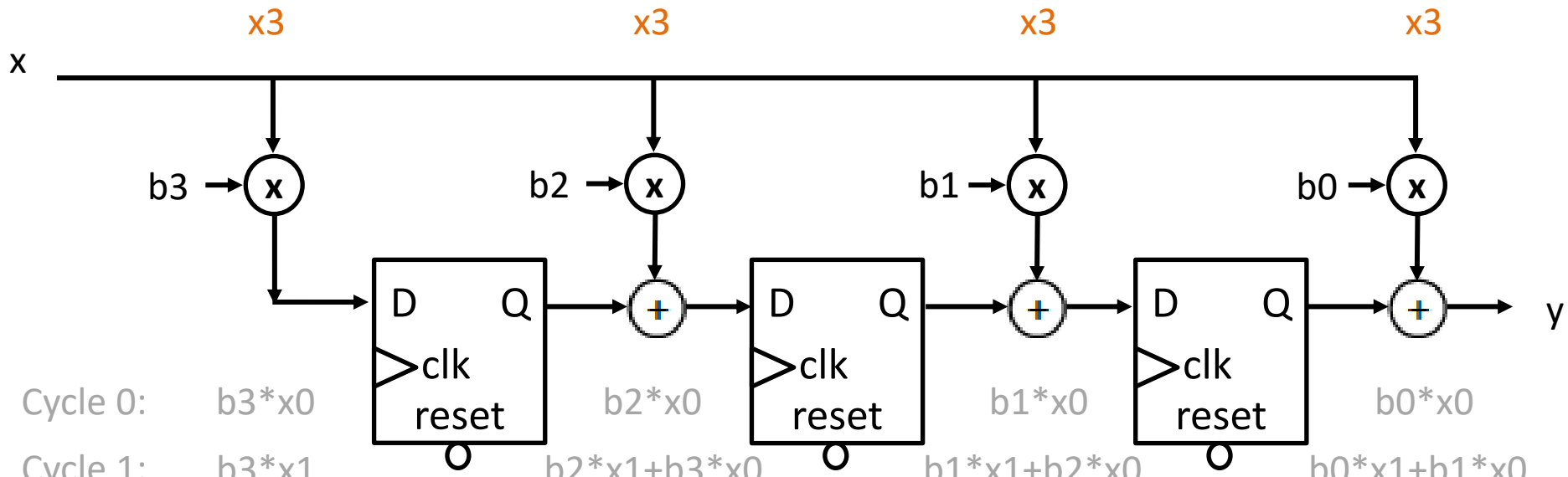
$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3]$$



Dataflow

- Cycle 3:

$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3]$$

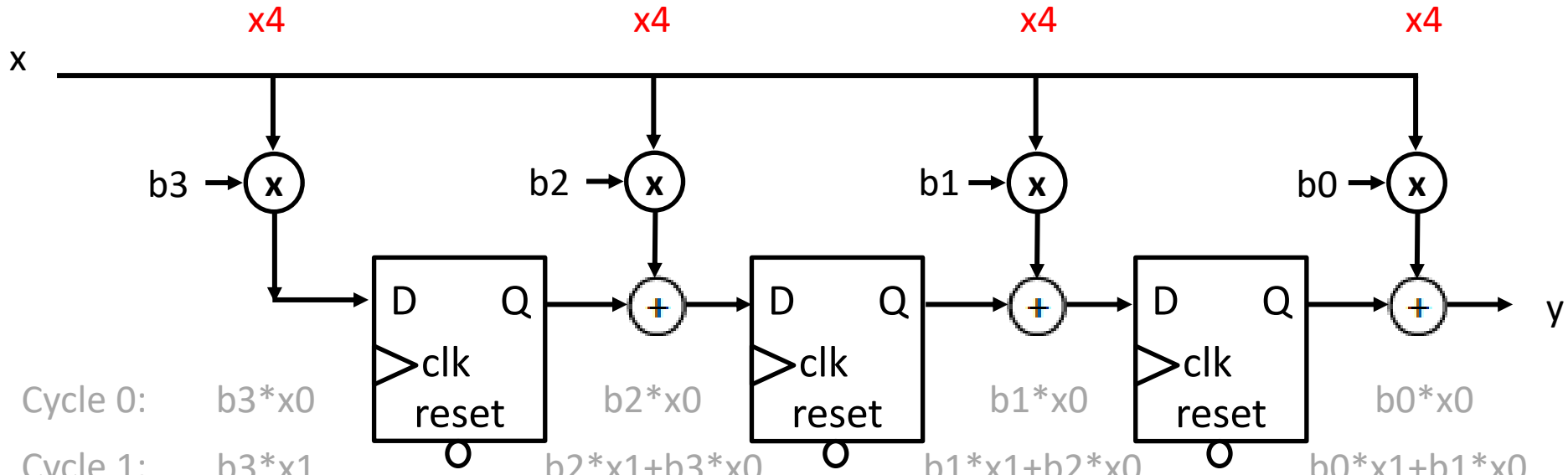


Cycle 0:	b3*x0		b2*x0	b1*x0	b0*x0
Cycle 1:	b3*x1		b2*x1+b3*x0	b1*x1+b2*x0	b0*x1+b1*x0
Cycle 2:	b3*x2		b2*x2+b3*x1	b1*x2+b2*x1+b3*x0	b0*x2+b1*x1+b2*x0
Cycle 3:	b3*x3		b2*x3+b3*x2	b1*x3+b2*x2+b3*x1	b0*x3+b1*x2+b2*x1+b3*x0 => y0

Dataflow

- Cycle 4:

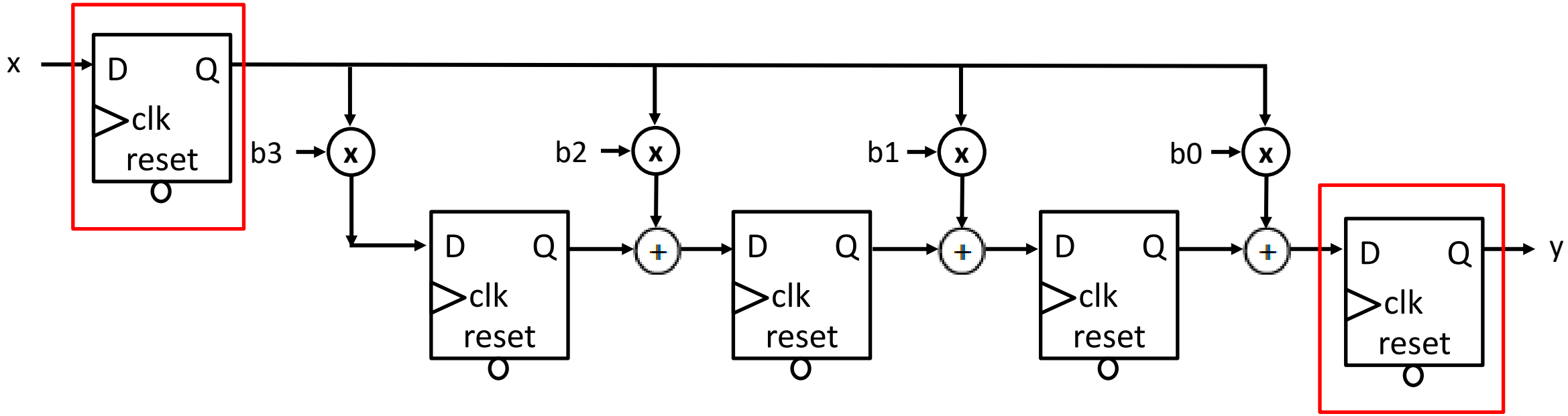
$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3]$$



Cycle 0:	b3*x0		b2*x0	b1*x0	b0*x0
Cycle 1:	b3*x1		b2*x1+b3*x0	b1*x1+b2*x0	b0*x1+b1*x0
Cycle 2:	b3*x2		b2*x2+b3*x1	b1*x2+b2*x1+b3*x0	b0*x2+b1*x1+b2*x0
Cycle 3:	b3*x3		b2*x3+b3*x2	b1*x3+b2*x2+b3*x1	b0*x3+b1*x2+b2*x1+b3*x0
Cycle 4:	b3*x4		b2*x4+b3*x3	b1*x4+b2*x3+b3*x2	b0*x4+b1*x3+b2*x2+b3*x1 => y1

Something you should know

- You need to block the input/output, or you will fail!



- critical path越長，合成時間越久。

FIR.sv

Input Signal	Bit Width	Definition
clk	1	Clock.
rst_n	1	Asynchronous active-low reset.
in_valid	1	High when input is valid.
weight_valid	1	High when weight is valid.
x	16	Unsigned input x.
b0, b1, b2, b3	16	Unsigned weight b0, b1, b2, b3.

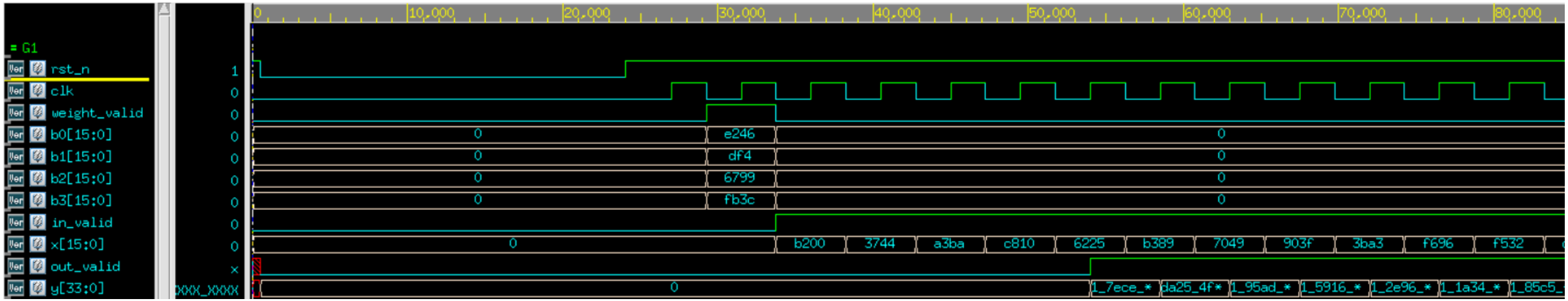
Output Signal	Bit Width	Definition
out_valid	1	High when the output is valid. Should be reset when rst_n is low. Should be continuous.
y	34	Unsigned output y. Should be reset when rst_n is low.

Specs

- 所有output必須asynchoronous negedge reset
- weight_valid上拉一個cycle後，in_valid連續上拉1003個cycle
- out_valid在第四個input x之後才能上拉
- out_valid必須連續
- out_valid必須在input 100個x之前上拉
- 連續輸出1000個正確答案即PASS
- Input/output signals皆為unsigned
- 01_RTL PASS
- 02_SYN clock period = 4.5ns, timing slack must be MET, no error and latch
- 03_GATE PASS, no error and timing violation

Output & Waveform

- Waveform



in_valid will be high for 1003 continuous cycles.
out_valid should be continuous.

Grading Policy

- Pass the RTL & Synthesis & Gate-level simulation: 100%
  - 合成結果**不能有Error、Timing report slack 必須是 MET、不能有Latch**
  - Gate-level simulation**不可以發生timing violation**
- Demo 2 打7折

```
Warning!  Timing violation
          $hold( posedge RN:119954 FS, posedge SN:204217 FS,  0.229292 : 229292 FS );
           File: /usr/cad/umc018/Verilog/umc18_neg.v, line = 9379
          Scope: TESTBED.I_Seq.lose_reg
           Time: 204217 FS
```

Upload

- 請將Lab09/01_RTL裡的FIR.sv依以下命名規則重新命名後上傳至E3
- 命名規則：FIR_dcsxxx.sv，xxx為工作站帳號號碼
- **命名錯誤扣5分!!!**
- **Deadline:**
  - **Demo 1: 5/8 17:25**
  - **Demo 2: 5/8 23:59**