Lab 6: Matrix Multiplication Circuit Design

National Chiao Tung University
Chun-Jen Tsai
11/4/2016

Lab 6: Matrix Multiplication

- In this lab, you will design a circuit to do 4×4 matrix multiplications.
  - The user press the WEST button to start the circuit
  - The circuit reads two 4×4 matrices from a file on the SD card, perform the multiplication, and print the output matrix through the UART to a terminal window
- You will demo the design to your TA during the lab hours on 11/22

Design Constraint of Lab6

- You must use no more than 16 multipliers to implement your circuit
  - Spartan-3E XC3S500E FPGA has 20 18-bit multipliers
- Your grade will be based on correctness, logic usage, performance, and your skill of coding

The Input Matrix Format

- Each input matrix has 16 unsigned 8-bit elements of values between 0 ~ 127 in column-major format

32 numbers for two 4×4 matrix

```
MATX_TAG↵
4F↵
7E↵
57↵
0F↵
14↵

…

7B↵
21↵
4C↵
54↵
```

- The output matrix has 16 unsigned 16-bit elements

The Output Format

- After the multiplication, the circuit prints the following messages to the UART:

```
The result is:↵
[ 5A2B, 33D6, 56C6, 5E75 ]↵
[ 64CF, 483E, 5225, 663D ]↵
[ 58D1, 4B9C, 5034, 5763 ]↵
[ 29D9, 3662, 2DA4, 24D3 ]↵
↵
```

Computation of $A_{4\times4} \times B_{4\times4}$

- A 4×4 matrix multiplication is composed of 16 inner products

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{pmatrix}$$

- In this lab, you can use either registers or SRAM to store the matrices

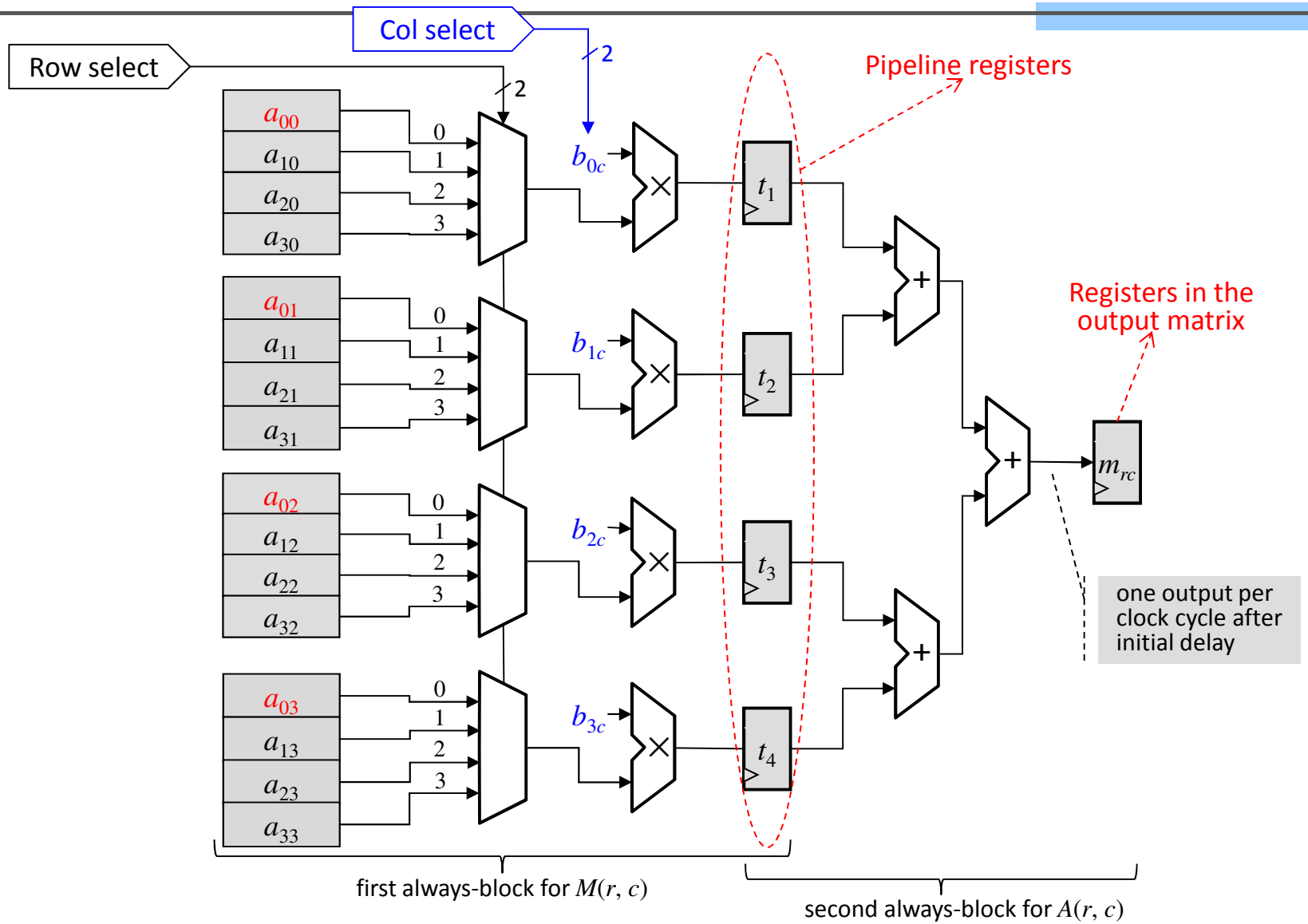
Pipeline Design for Data Processing

- The computation of the inner product of row r and column c can be divided into two steps:
  - $M(r,c)$: parallel computation of $t_j = a_{rj} \times b_{jc}$, $j = 0, 1, 2, 3$
  - $A(r,c)$: compute adder-tree of $(t_0 + t_1) + (t_2 + t_3)$

- These two circuit blocks $M(r,c)$ and $A(r,c)$ can be arranged into a processing pipeline:

Data \ Clock	0	1	2	3	4
$r = 0, c = 0$	$M(0, 0)$	$A(0, 0)$			
$r = 1, c = 0$		$M(1, 0)$	$A(1, 0)$		
$r = 2, c = 0$			$M(2, 0)$	$A(2, 0)$	
$r = 3, c = 0$				$M(3, 0)$	$A(3, 0)$

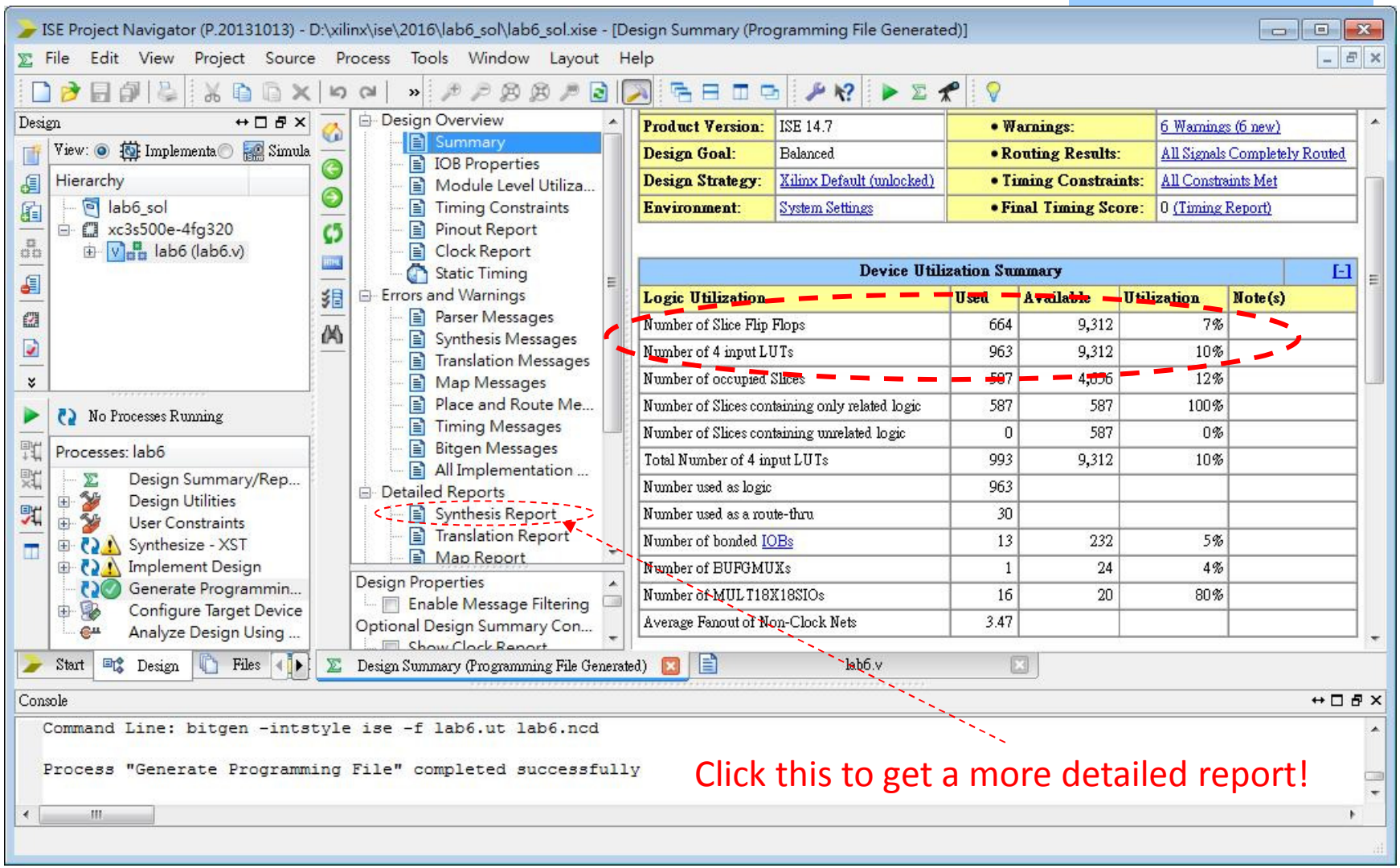
Possible Datapath of $A_{4\times4} \times b_{4\times1}$



Design a FSM to Re-use the Datapath

- You can design an FSM to control the row select signal so that the $A_{4\times4} \times b_{4\times1}$ datapath can finish computation in 4+1 clock cycles (note that there is a 1-cycle delay)
- For $A_{4\times4} \times B_{4\times4}$, you can duplicate the matrix-vector multiplication module four times, so, 4×4 matrix multiplication can be computed in 5 clock cycles
  - This is quite costly in circuit resource
- As an alternative, you can reuse the datapath in page 7 for each column of B matrix. It takes 16 cycles to finish the multiplication
  - This approach saves a lot of circuit resource

Resource Usage Summary



Synthesis Report

- Synthesis report provides a detailed analysis of your circuit and show some timing information.

```
=====================================
HDL Synthesis Report

Macro Statistics
# Multipliers                       : 16
 8x8-bit multiplier                 : 16
# Adders/Subtractors                : 23
 11-bit subtractor                  : 2
 16-bit adder carry out             : 8
 17-bit adder carry out             : 4
 4-bit subtractor                   : 2
 6-bit subtractor                   : 2
 8-bit adder                        : 5
# Counters                          : 5
 2-bit up counter                   : 1

 . . . . .

 5-bit comparator less              : 1
# Multiplexers                      : 6
 18-bit 16-to-1 multiplexer         : 1
 8-bit 4-to-1 multiplexer           : 4
 8-bit 8-to-1 multiplexer           : 1

=====================================
```

```
Device utilization summary:
---------------------------

Selected Device : 3s500efg320-4

 Number of Slices:              590  out of   4656    12%
 Number of Slice Flip Flops:    664  out of   9312     7%
 Number of 4 input LUTs:        985  out of   9312    10%
 Number of IOs:                  13
 Number of bonded IOBs:          13  out of    232     5%
 Number of MULT18X18SIOs:        16  out of     20    80%
 Number of GCLKs:                 1  out of     24     4%

  . . . . .


Timing Summary:
---------------
Speed Grade: -4

 Minimum period: 11.792ns (Maximum Frequency: 84.803MHz)
 Minimum input arrival time before clock: 11.348ns
 Maximum output required time after clock: 6.145ns
 Maximum combinational path delay: No path found
```