Outline

10.2 Streaming Design on DDOT

10.3 Iterative Design with Streaming Width Four

10.2 Streaming Design on DDOT

- 10.2.1 Design Structure of Streaming Design on DDOT
  - The streaming width is configured at eight, enabling the input of eight FP data sets ``x0-x7'' and eight FP data sets ``y0-y7'' into the design engine during each clock cycle.
  - The number of clock cycles needed to feed each data frame into the design engine:

$$NO\_CY = \lceil N/STW \rceil = \lceil 8/8 \rceil = 1.$$

  - The number of timing controllers needed:

$$NO\_TCs = \lceil Longest\_Path/NO\_CY \rceil = \lceil 4/1 \rceil = 4.$$

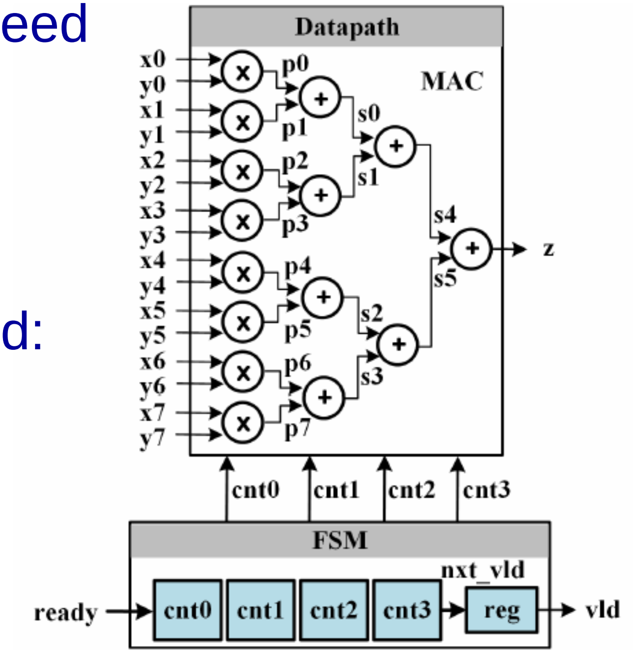
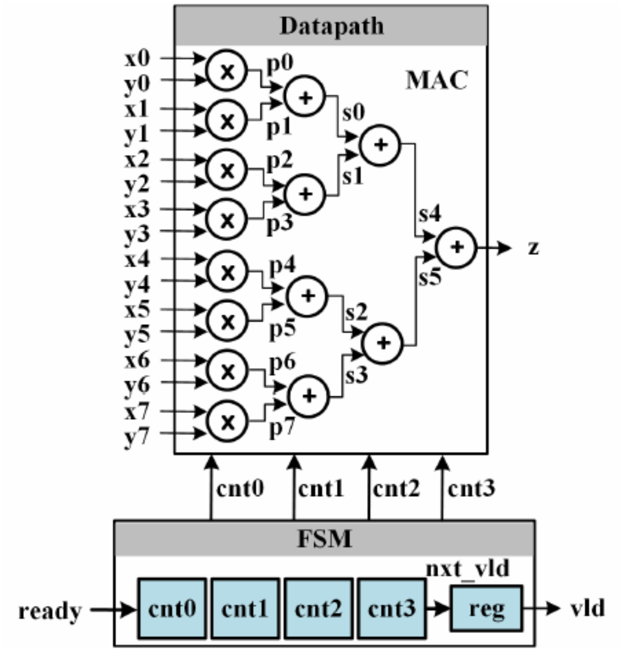


FIGURE 10.2
Streaming Design on DDOT

10.2 Streaming Design on DDOT

- 10.2.2 Timing Diagram of Streaming Design on DDOT
  - The counter labeled ``cnt0'' serves to monitor the processing of the first data frame across clock cycles 1-4.
  - The other three counters, namely ``cnt1'', ``cnt2'', and ``cnt3'', oversee the ddot processing of the subsequent three data frames.
  - 1st package: 8*0.0 and 8*1.0, 2nd package: 8*1.0 and 8*2.0, until the eighth package: 8*7.0 and 8*8.0



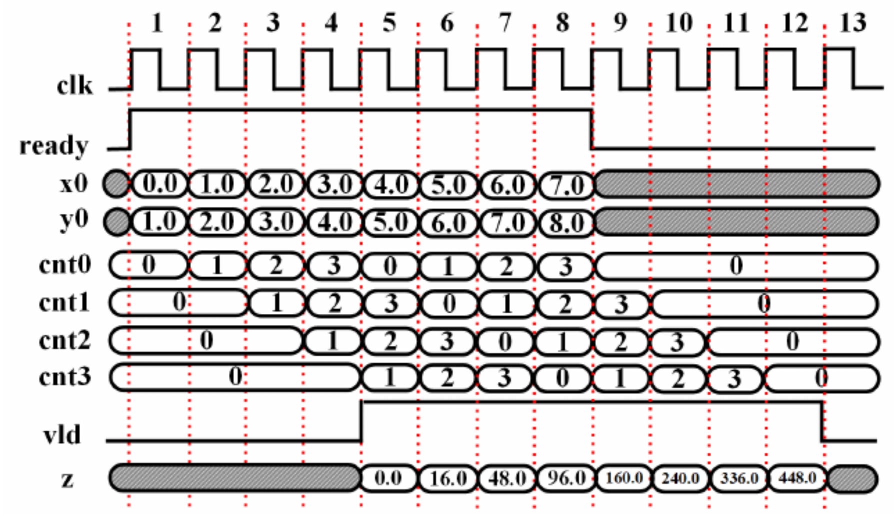


FIGURE 10.3
Timing Diagram of Streaming Design on DDOT

10.2 Streaming Design on DDOT

- 10.2.3 Timing Controller of Streaming Design on DDOT
  - The Overall Architecture of Each Timing Controller
    - A sequential counter
    - A counter enable circuit
      - The signals labeled with ``x'' in the names, such as ``stx_rdy”, ``cntx_en”, ``nxt_cntx”, and ``cntx”, correspond to the control signals for all the counters: ``cnt0'' to ``cnt3”.
      - The ``cntx_en'' output is activated by the ``stx_rdy'' pulse signal and stays active as long as the counter is actively monitoring data. This condition is determined through the use of a Reduction OR gate.
      - The ``stx_rdy'' trigger signal is controlled by the arbiter, a critical component responsible for ensuring the desired rotation sequence of counters within the counter enable circuit.

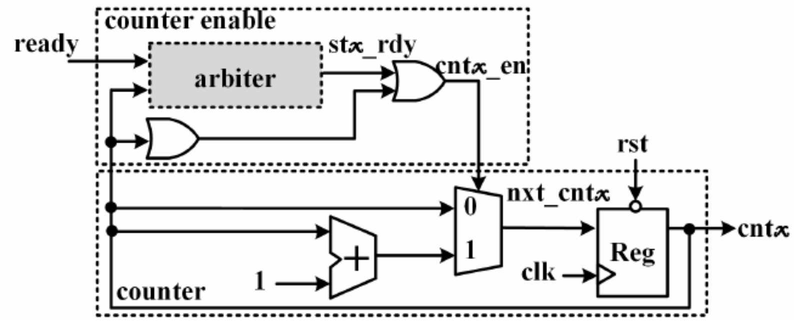
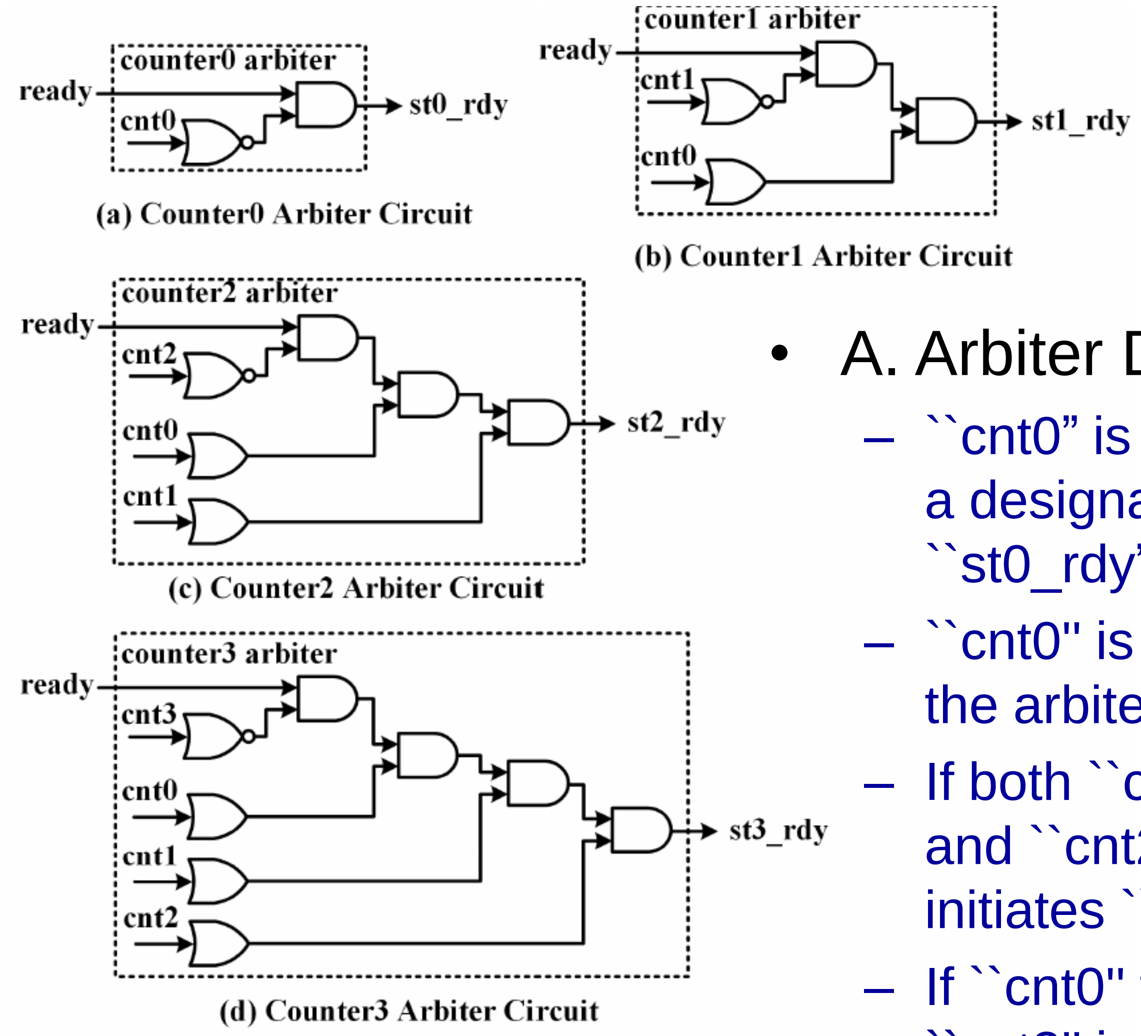


FIGURE 10.4
Timing Controller of Streaming Design on DDOT

- 10.2.3 Timing Controller of Streaming Design on DDOT
  - Arbiter Design
    - The primary challenge lies in coordinating the rotation of these four controllers in a predetermined order, based on dynamic statuses.
    - The pre-defined priorities of these four counters, from highest to lowest, are labeled as ``cnt0'', ``cnt1'', ``cnt2'', and ``cnt3”.
  - Arbiter Design Methodology
    - The ``cnt0” will be employed as long as it is available;
    - If ``cnt0” is busy and the ``cnt1” is available, then ``cnt1” will be selected;
    - If both ``cnt0” and ``cnt1” are busy, and ``cnt2” is ``cnt1”, then ``cnt2” will be selected;
    - If all ``cnt0”, ``cnt1”, and ``cnt2” are busy, and ``cnt3” is available, then ``cnt3” will be selected.
    - *Busy (non-zero): Reduction OR;*
    - *Available (zero): NOT reduction OR*

10.2 Streaming Design on DDOT



(a) Counter0 Arbiter Circuit

(b) Counter1 Arbiter Circuit

(c) Counter2 Arbiter Circuit

(d) Counter3 Arbiter Circuit

FIGURE 10.5
Arbiter Design of Timing Controller (Streaming Design on DDOT)

- A. Arbiter Design
  - ``cnt0” is available, it's activated using a designated pulse signal labeled as ``st0_rdy”.
  - ``cnt0" is busy and ``cnt1" is available, the arbiter triggers ``st1_rdy”.
  - If both ``cnt0" and ``cnt1" are busy, and ``cnt2" is available, the arbiter initiates ``st2_rdy”.
  - If ``cnt0" to ``cnt2" are busy, but ``cnt3" is available, the arbiter commences ``st3_rdy".

10.2 Streaming Design on DDOT

- B. Timing Controllers
  - Commencing with the initial data frame, control is assumed by the ``cnt0'' spanning clock cycles 1 to 4.
  - The second data frame regulated by ``cnt1'', operating from clock cycles 2 to 5, then the third data frame guided by ``cnt2'' spanning clock cycles 3 to 6. Subsequently, the fourth data frame is overseen by ``cnt3'' over clock cycles 4 to 7.
  - It follows with another four data frames over clock cycles 5-8, being processed by ``cnt0-cnt3'' sequentially.

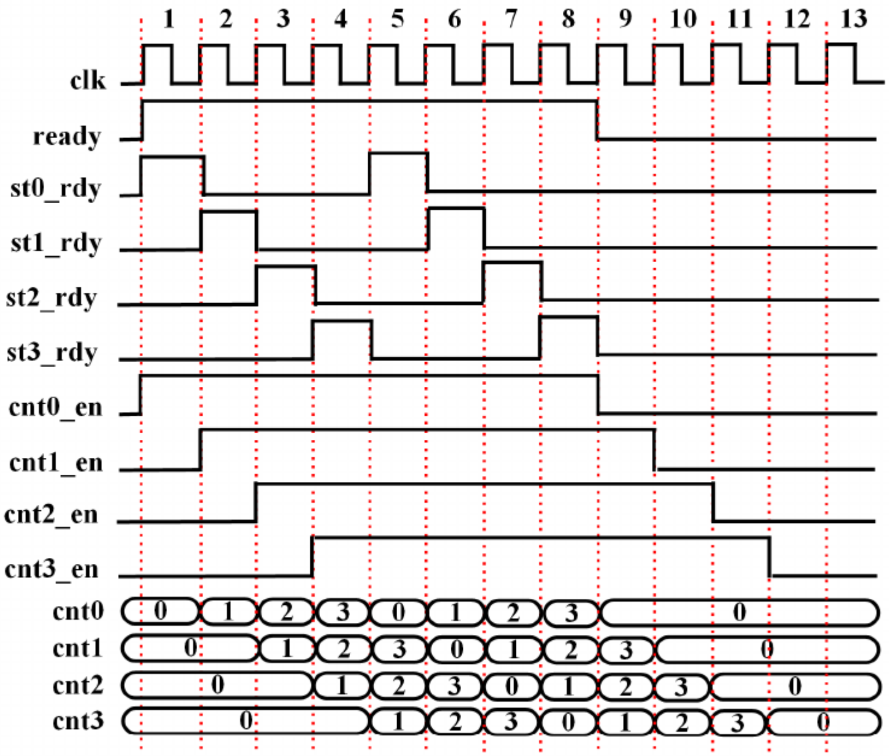
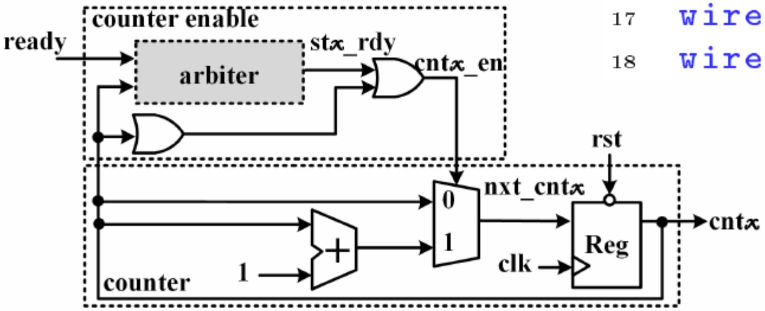


FIGURE 10.6
Timing Diagram of Timing Controller (Streaming Design on DDOT)

10.2 Streaming Design on DDOT

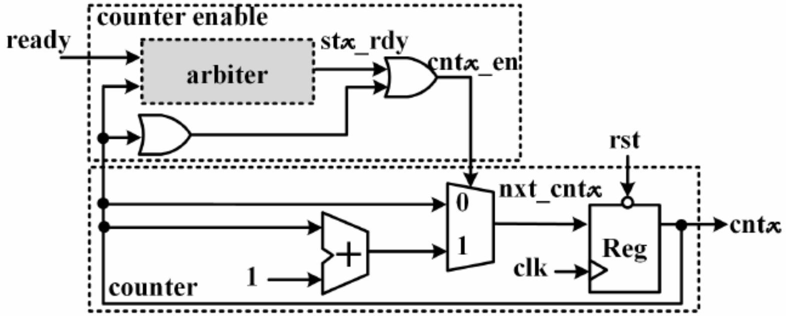
- 10.2.4 Verilog Code for Streaming Design on DDOT
  - Timing controller design

```
//****************************************
//**** Timing Controller Design *********
//****************************************
reg  [1:0] cnt0, cnt1, cnt2, cnt3;
wire st0_rdy = ready & ~|cnt0 ;
wire st1_rdy = ready & ~|cnt1 & |cnt0 ;
wire st2_rdy = ready & ~|cnt2 & |cnt0 & |cnt1 ;
wire st3_rdy = ready & ~|cnt3 & |cnt0 & |cnt1 & |cnt2 ;

wire cnt0_en = st0_rdy | |cnt0 ;
wire cnt1_en = st1_rdy | |cnt1 ;
wire cnt2_en = st2_rdy | |cnt2 ;
wire cnt3_en = st3_rdy | |cnt3 ;

wire [1:0] nxt_cnt0 = cnt0_en ? (cnt0+2'd1) : cnt0;
wire [1:0] nxt_cnt1 = cnt1_en ? (cnt1+2'd1) : cnt1;
wire [1:0] nxt_cnt2 = cnt2_en ? (cnt2+2'd1) : cnt2;
wire [1:0] nxt_cnt3 = cnt3_en ? (cnt3+2'd1) : cnt3;
```



- 10.2.4 Verilog Code for Streaming Design on DDOT
  - Timing controller design

```
always @(posedge clk) begin
  if(rst) begin
    cnt0<=2'd0    ;
    cnt1<=2'd0    ;
    cnt2<=2'd0    ;
    cnt3<=2'd0    ;
  end else begin
    cnt0<=nxt_cnt0;
    cnt1<=nxt_cnt1;
    cnt2<=nxt_cnt2;
    cnt3<=nxt_cnt3;
  end
end

wire nxt_vld = &cnt0 | &cnt1 | &cnt2 | &cnt3;
always @(posedge clk) begin
  if(rst) begin
    vld  <=1'b0    ;
  end else begin
    vld  <=nxt_vld ;
  end
end
```



Outline

10.2 Streaming Design on DDOT

10.3 Iterative Design with Streaming Width Four

10.3 Iterative Design with Streaming Width Four

- Iterative Design vs. streaming design
  - Streaming design
    - Features eight sets of single-precision FP inputs
    - Hardware cost:
      - 512 single-bit IO connections (resulting from 16x32 connections), along with the incorporation of eight FP multipliers and seven FP adders.
  - Iterative Design
    - Strike a balance between resource utilization and meeting the specified requirements
    - Enabled a reduction in the streaming width within the design engine, thereby decreasing the number of necessary IO connections and overall resource costs.

10.3 Iterative Design with Streaming Width Four

- 10.3.1 Iterative Design Structure with Streaming Width Four
  - Employs a streaming width of four, enabling the input of a complete data frame, comprising eight FP data, within two clock cycles.
  - The number of clock cycles needed to feed

each data frame into the design engine:

$$NO\_CY = \lceil N/STW \rceil = \lceil 8/4 \rceil = 2.$$

  - The number of timing controllers needed:

$$NO\_TC = \lceil Longest\_Path/NO\_CY \rceil = \lceil 5/2 \rceil = 3.$$

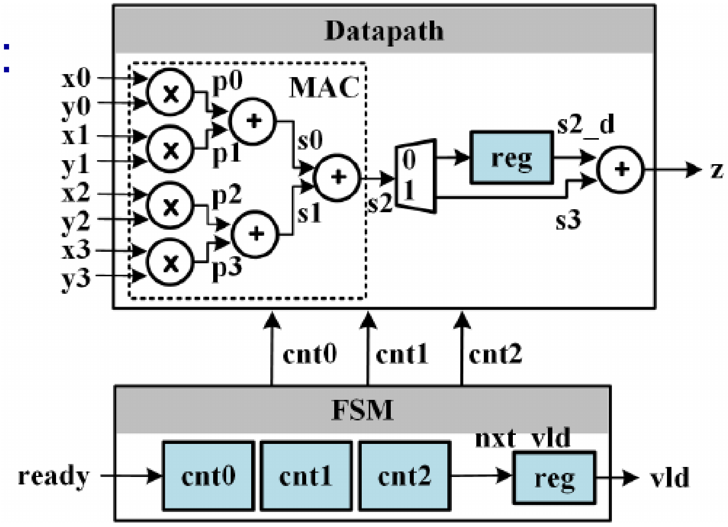


FIGURE 10.7
Iterative Design Structure with Streaming Width Four

10.3 Iterative Design with Streaming Width Four

- 10.3.1 Iterative Design Structure with Streaming Width Four
  - Data path design
    - Multiplication-Addition Circuit (MAC)
      - Four FP multipliers and three consecutive cascading FP adders
    - DeMux
      - The first MAC output is routed to a register output
      - The second MAC output is routed to

      The input of the following FP adder

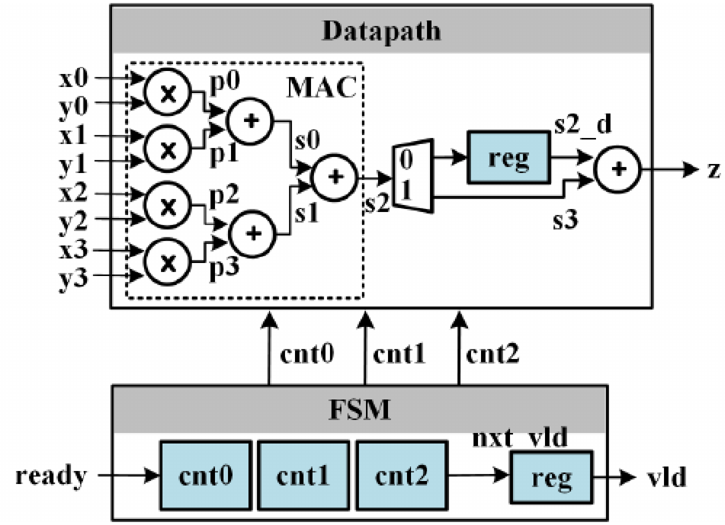
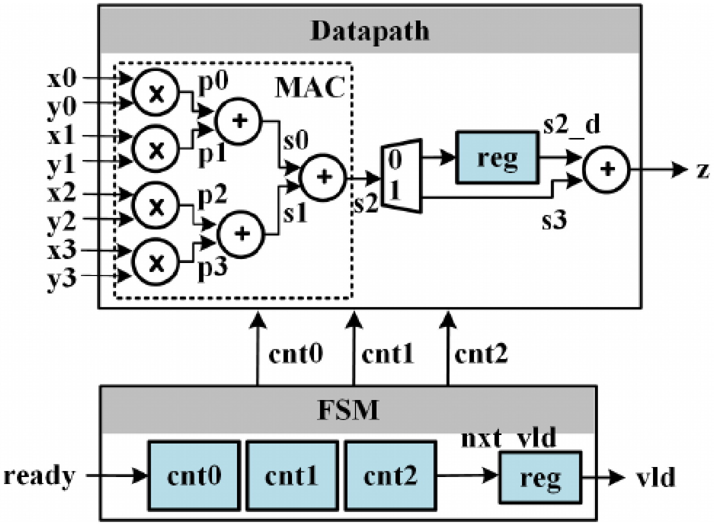


FIGURE 10.7
Iterative Design Structure with Streaming Width Four

10.3 Iterative Design with Streaming Width Four

- 10.3.2 Timing Diagram of Iterative Design with Streaming Width Four
  - Four Data Frames Example
    - The products ``p0-p3'' can be obtained in the second clock cycle, the summations ``s0-s1'' can be reached by the third clock cycle, and the intermediate MAC result ``s2'' for the first half data frame can be received by the fourth clock cycle.
    - Subsequently, this intermediate outcome is registered as ``s2_d'' within the fifth clock cycle.



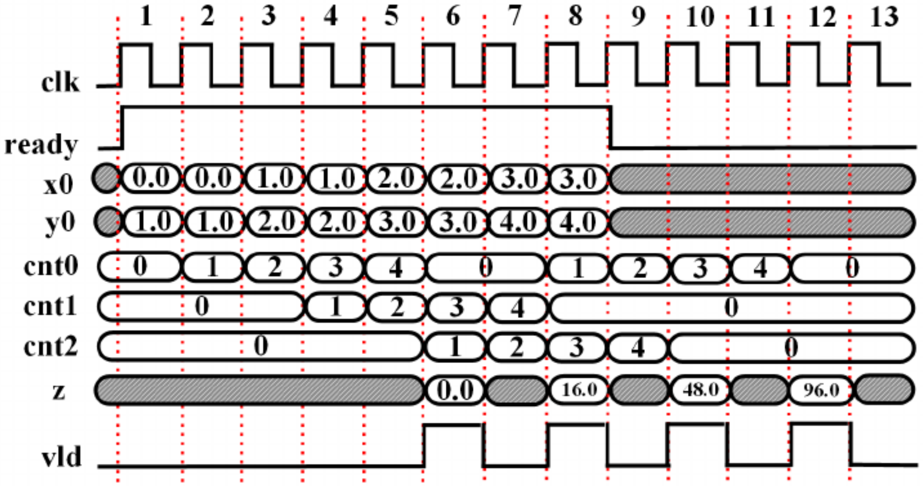
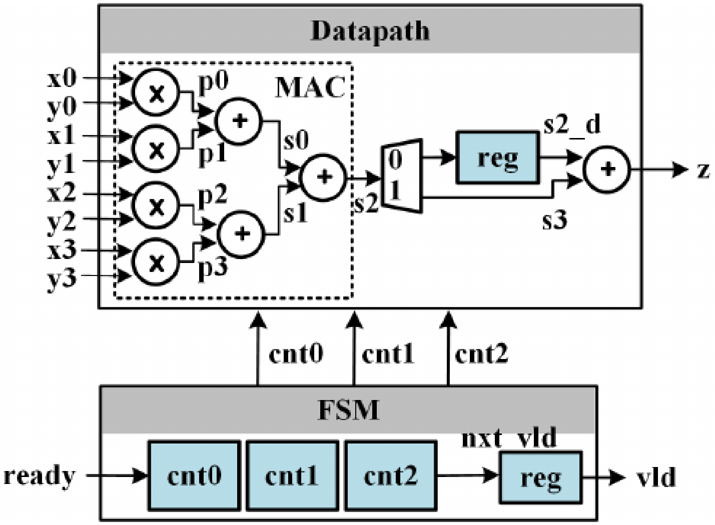


FIGURE 10.8
Timing Diagram of Iterative Design with Streaming Width Four

10.3 Iterative Design with Streaming Width Four

- 10.3.2 Timing Diagram of Iterative Design with Streaming Width Four
  - By the fifth clock cycle, the MAC outcome ``s2'' for the second half of the same data frame also comes to the fore, subsequently being channeled to the ``s3'' bus. Significantly, the ultimate outcome can be achieved by accumulating ``s2_d'' and ``s3'', effectuating the final summation ``z = FP 0.0'' within the sixth clock cycle.
  - Simultaneously, the counters ``cnt1'' and ``cnt2'' contribute to the concurrent management of the second and third data frames.



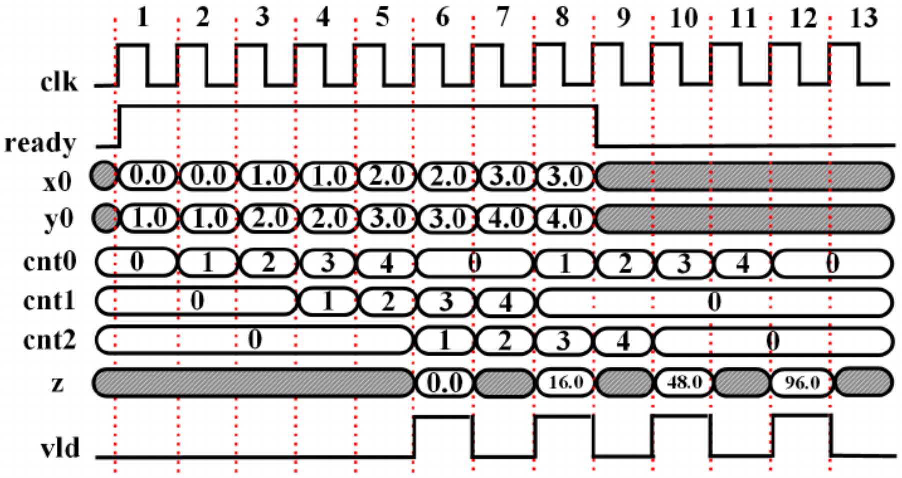
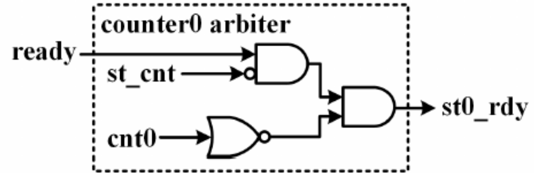


FIGURE 10.8
Timing Diagram of Iterative Design with Streaming Width Four

10.3 Iterative Design with Streaming Width Four

- 10.3. 3 Timing Controller of Iterative Design with Streaming Width Four
  - ``st_cnt'' counter
    - It is instrumental in distinguishing these two clock cycles: the first clock cycle is marked when ``st_cnt'' is zero, leading to the use of an inverter output as the input of the AND gate.
  - ``st0_rdy'' start indicator
    - is exclusively allowed during the first clock cycle of each data frame input period, and only when ``cnt0'' is in an idle state.



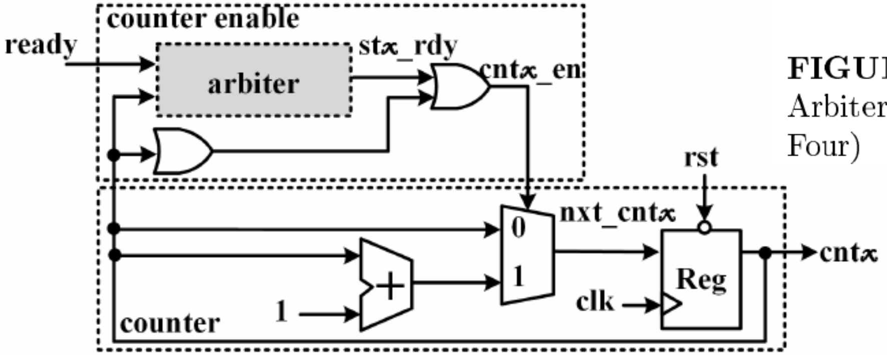


FIGURE 10.9
Arbiter Design of Timing Controller (Iterative Design with Streaming Width Four)

10.3 Iterative Design with Streaming Width Four

- 10.3. 3 Timing Controller of Iterative Design with Streaming Width Four
  - The ``st_cnt'' counter is utilized to keep track of the 2-clock cycle duration of each data frame input.
  - The counter enable signals ``cnt0_en‘’, ``cnt1_en'', and ``cnt2_en‘’ are triggered the moment the corresponding start indicator is active.

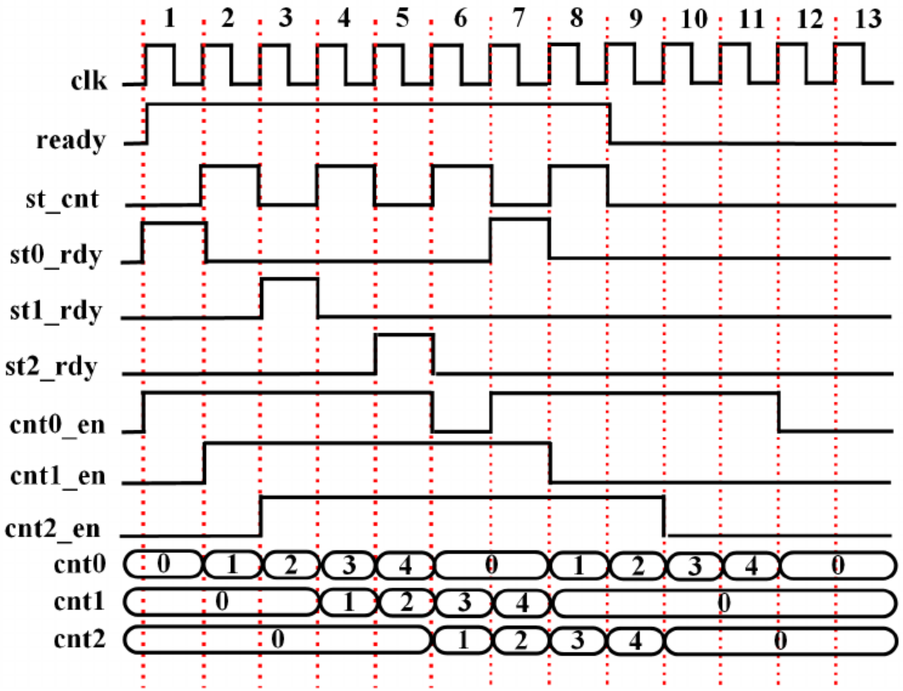


FIGURE 10.10
Timing Diagram of Timing Controller (Iterative Design with Streaming Width Four)