REPORT

DCS HW4
Student: 111511076 陳彥宇

Design—Operation During Each Clk Cycle:

This assignment doesn't really require a state graph. I figured that a chart detailing the actions performed during each cycle is far more useful in clarifying my design.

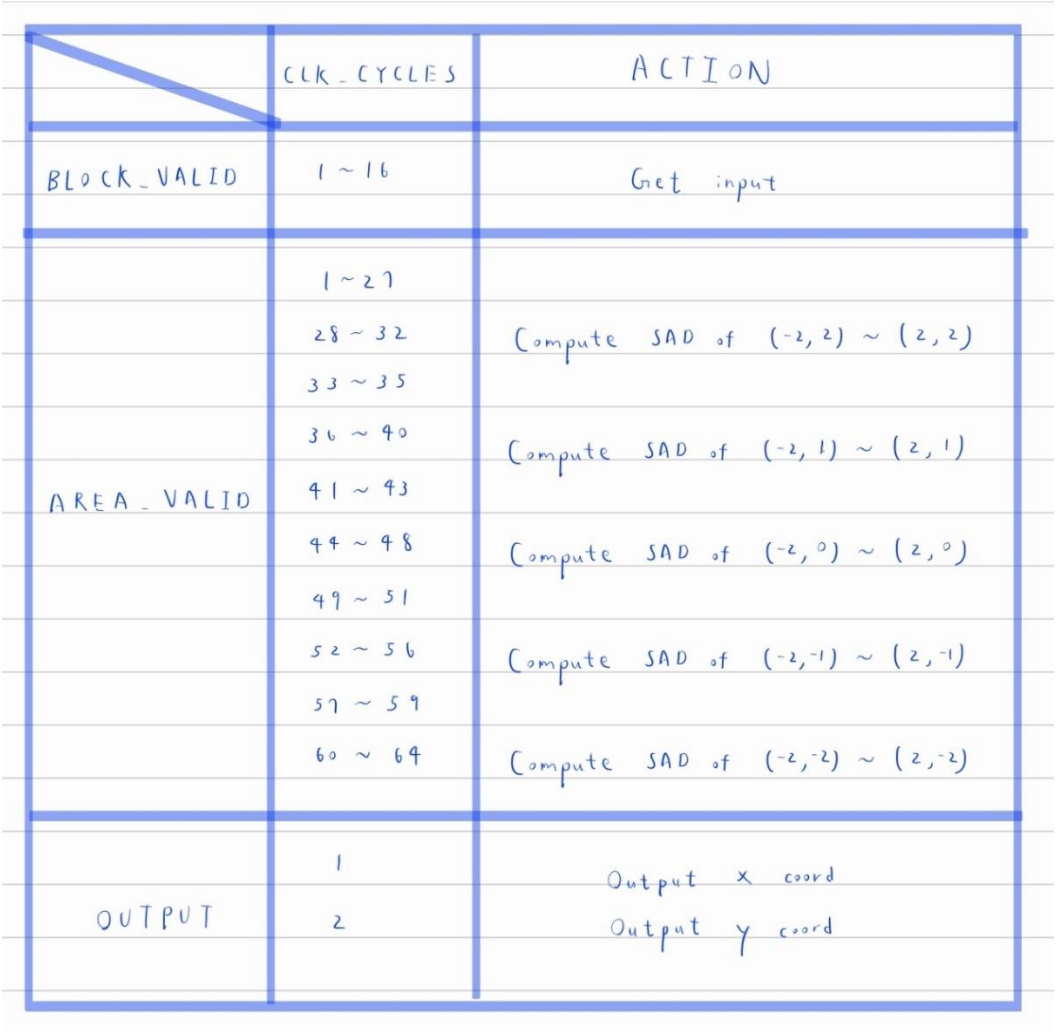


Fig1: Timing Chart

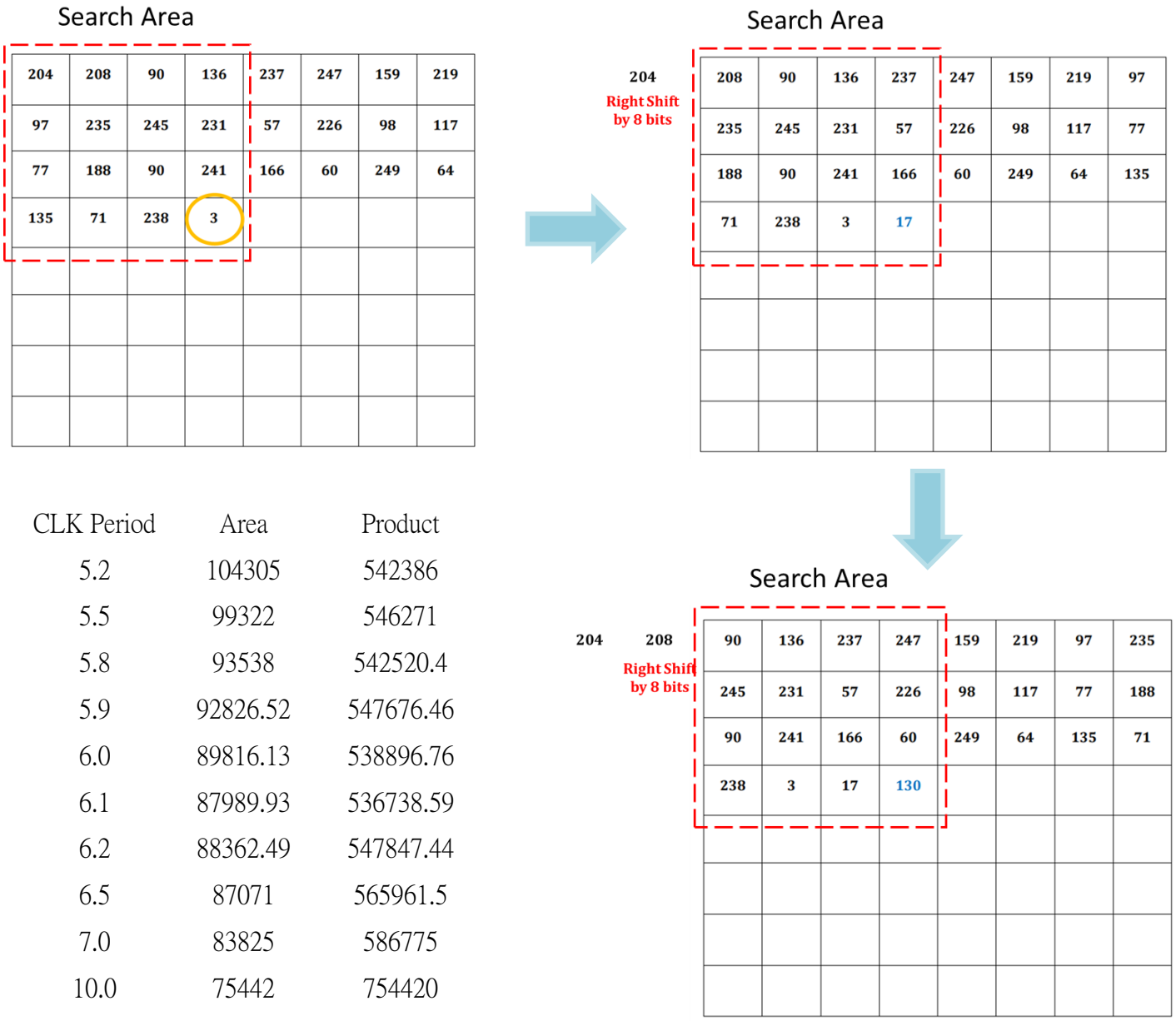
Area & Timing Optimization:

Initially, I tried to move the search area across the whole 8 x 8 area, but I soon encountered difficulties. I realized that it's illegal to access elements in an array like so, arr[i : i-3] using a non-constant i. Then, I figured, "Why not do it the other way around?" it's much easier to shift the numbers into the search area than moving the search area itself. Therefore, I used the following method to compute the SAD at each position.

At first, I set indexes i and j to move across the array normally. Nothing special about that. However, once the position(3,3) is reached, my indexes i & j are locked. Starting from this point, I start shifting the entire packed array to the left by 8 bits every cycle. (Please refer to the following figure) This will remove the number at (0,0) and make room for the next in_data at (3,3). Notice that this allows us to compute the SAD at (-2,2), (-1,2), (0,2) and so on while parsing inputs. However, an important thing to note is that when the x coordinate reaches 2, (Positions (2,2), (2,1), … (2,-2)) we must wait 3 cycles WITHOUT

computing the SAD. Once three cycles have elapsed and three new inputs are in positions (3,1), (3,2), (3,3) respectively, we can continue the computations. Since it's not that easy to illustrate the reason behind this in the report, one can try shifting the numbers or looking at the waveforms themselves to see why this is the case.

Fig2: Shifting Process



CLK Period	Area	Product
5.2	104305	542386
5.5	99322	546271
5.8	93538	542520.4
5.9	92826.52	547676.46
6.0	89816.13	538896.76
6.1	87989.93	536738.59
6.2	88362.49	547847.44
6.5	87071	565961.5
7.0	83825	586775
10.0	75442	754420

Fig3: Clk Period & Area Product

The minimum clk period that my design can utilize is 5.2ns. However, notice from the chart above that this is not the most optimal choice. Reducing the clk period will force the synthesizer to increase the area of the design. After checking through different clk period, I found that 6.1ns will allow me to achieve the best balance between timing and area.

Difference Between Packed and Unpacked Array:

- ✧ Unpacked arrays are stored in separate memory locations, with each element occupying its own memory space.
- ✧ All elements of a packed array are stored in consecutive memory locations. My shifting method mentioned above requires the use of packed arrays.

```
reg [7:0] unpacked_arr [0:3][0:3];
reg [7:0][0:3][0:3] packed_arr;
```