Design FFT Processor

Memory Architecture

By:

Hamzah Ahmed Ali

Arabic computer engineering community

http://www.arabcec.com/

20 August 2008

Memory Based Architecture is one of the famous architectures used in desgn Fast Fourier Transform fft processor, the basic idea of this architecture is to store the input point in memory component then perform the following processing cycle (catch data, calculate, and save result in memory again). As we see in figure 1, this architecture contain two basic elements; Memory/Ram component that contain the input point data, interleaving calculated data while processing, and the output point data, The second component is the processing unit , this element just contain butterfly processor element(s). The number of butterflies elements or Memory/Ram banks in this architecture is depending on the design issues and specifications. We can improve the performance of this architecture by adding an additional cache memory between memories and processing block to increase the efficiency of memory accessing.

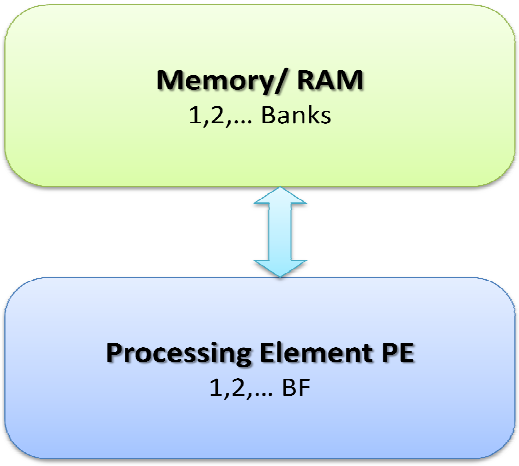


Figure 1: Memory Based Architecture

The following section will describe a very simple design of FFT processor using memory based architecture, to make this architecture more clearly to reader.

Simple memory architecture design

This design consists of one memory bank and one butterfly processor unit, as we see in figure 2. By looking to this design we note that, it consists of simple component and can be implemented in a simple fashion. Indeed, it can fitting in more small area for long FFT application and save more power, but the main disadvantage of it is low throughput thus make it inapplicable in communication industry. Also memory access is very simple, but we must wait two memory access cycles to start to calculate the butterfly operation and wait another two cycles to save the result in memory

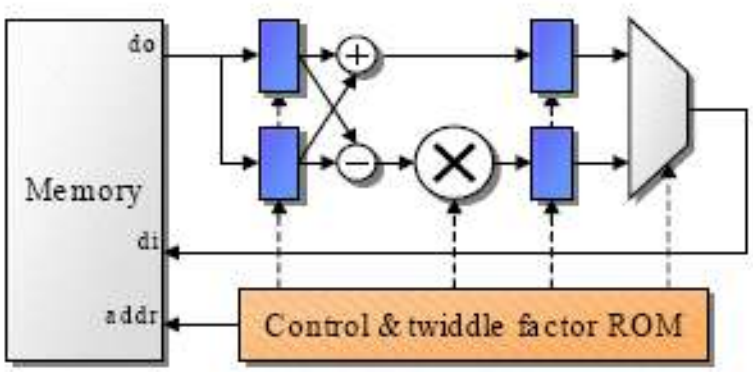


Figure 2: Simple design of FFT processor using memory architecture

Memory partitions

To improve the above design we need to divide the memory into more than one bank, figure 3 show the different between two design, the first use one memory bank and the second use two memory banks.

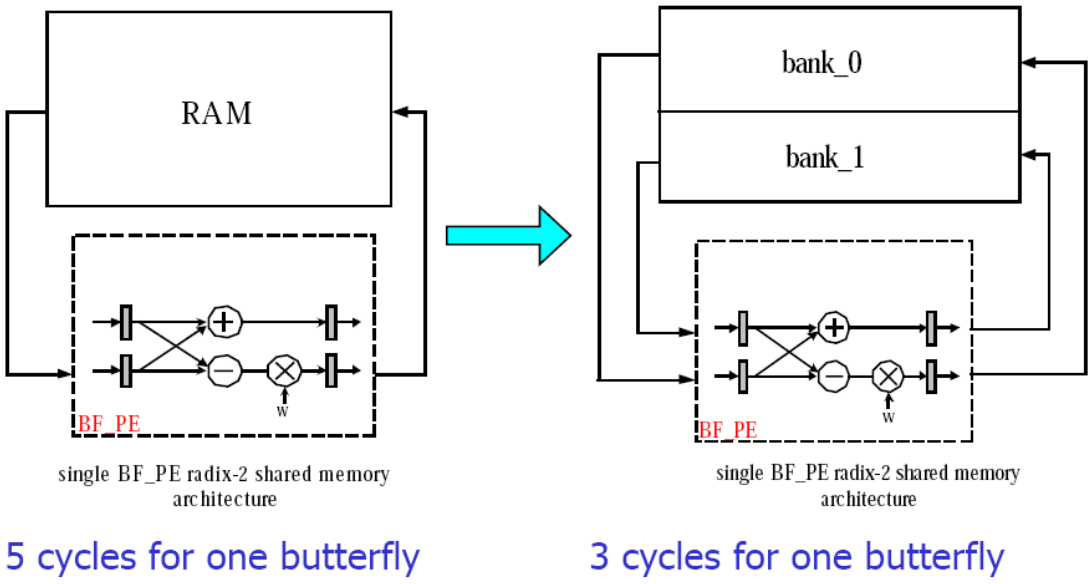
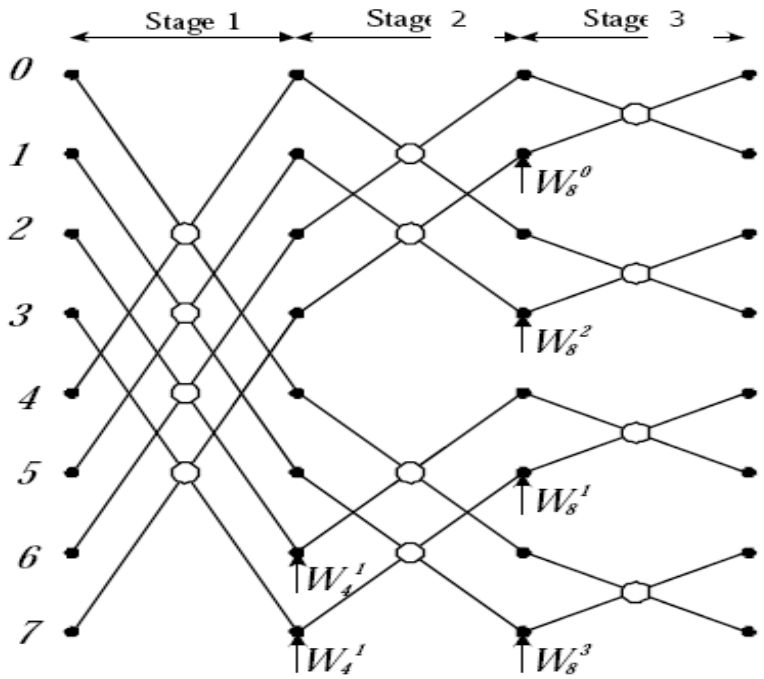


Figure 3: Enhancement memory architecture by partition memory into more than one block

As we see in one bank design we need 5 cycles to perform one butterfly processing, but when adding only one additional bank, it will be reduced to 3 cycles.

But which point will store in the bank0 and bandk1, because we need to store points of each pair of point –points that will apply to them the butterfly operations- in different banks. Figure 4 show an example of memory partition problem in 8 FFT.



Stage 1		Stage 2		Stage 3	
0	4	0	2	0	1
1	5	4	6	2	3
2	6	1	3	4	5
3	7	5	7	6	7

Figure 4: Data access pattern of 8-point FFT at different stages with two disjoint sets

We can solve this problem easily by applying coloring graph algorithm, as shown in figure 5.

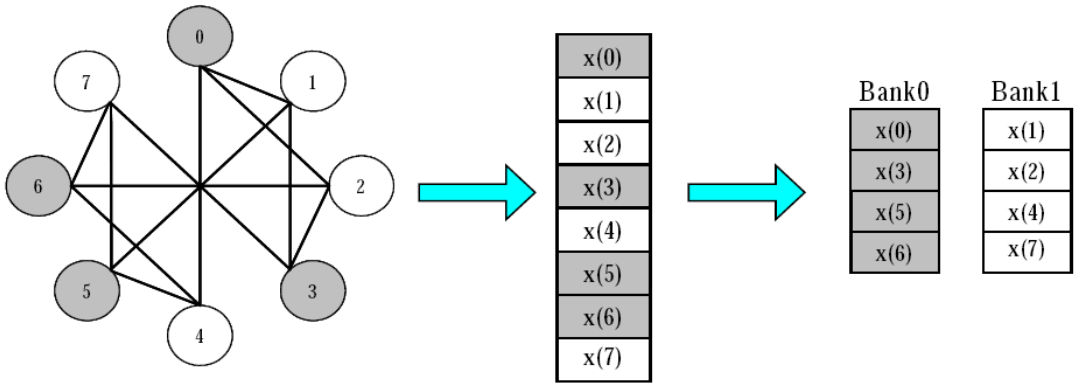


Figure 5: Appling coloring graphic algorithm to solve memory partition problem