Transform Processing on a Reconfigurable Data Path Processor

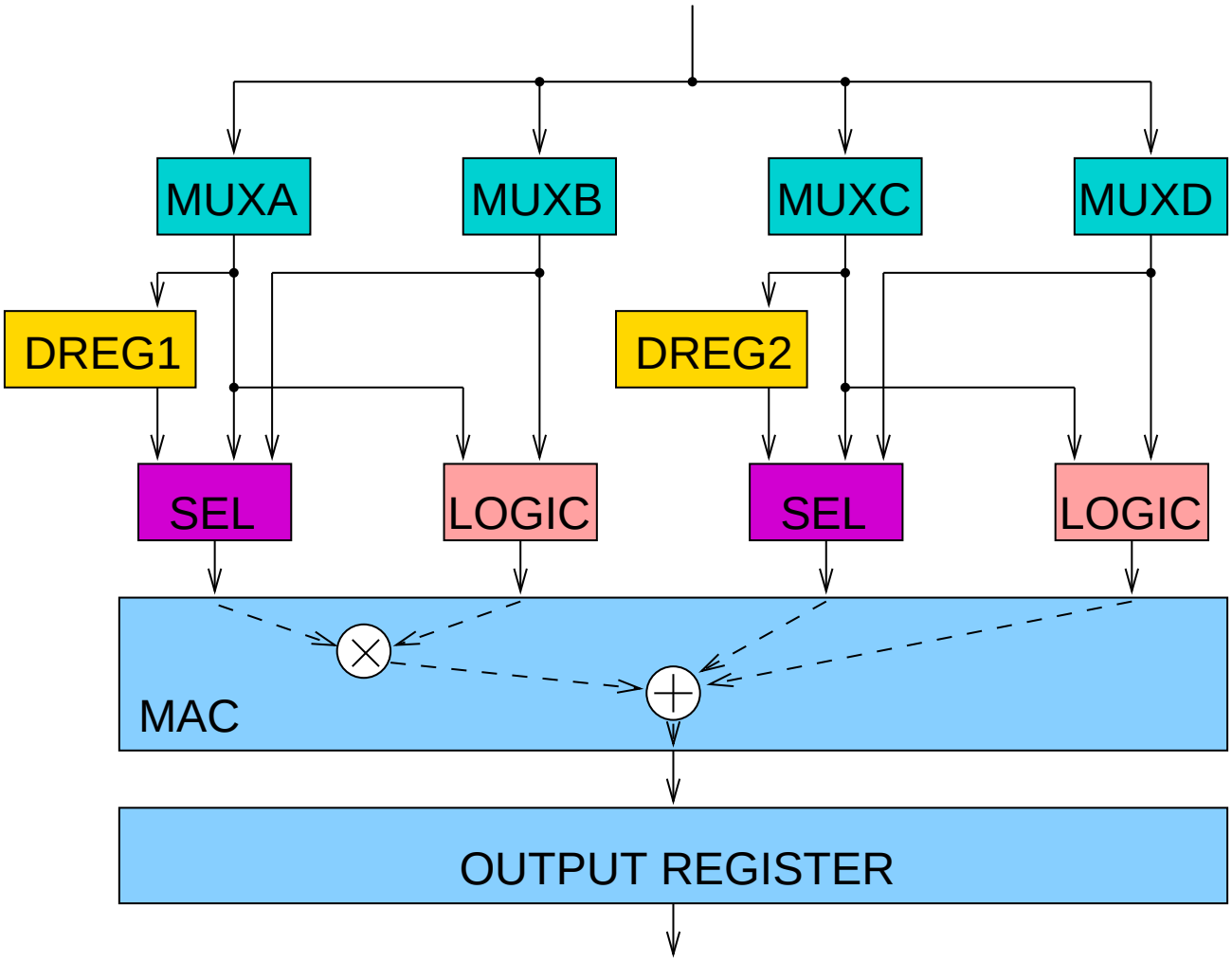
Joe Hass and Dave Cox

Institute of Advanced Microelectronics

The University of New Mexico

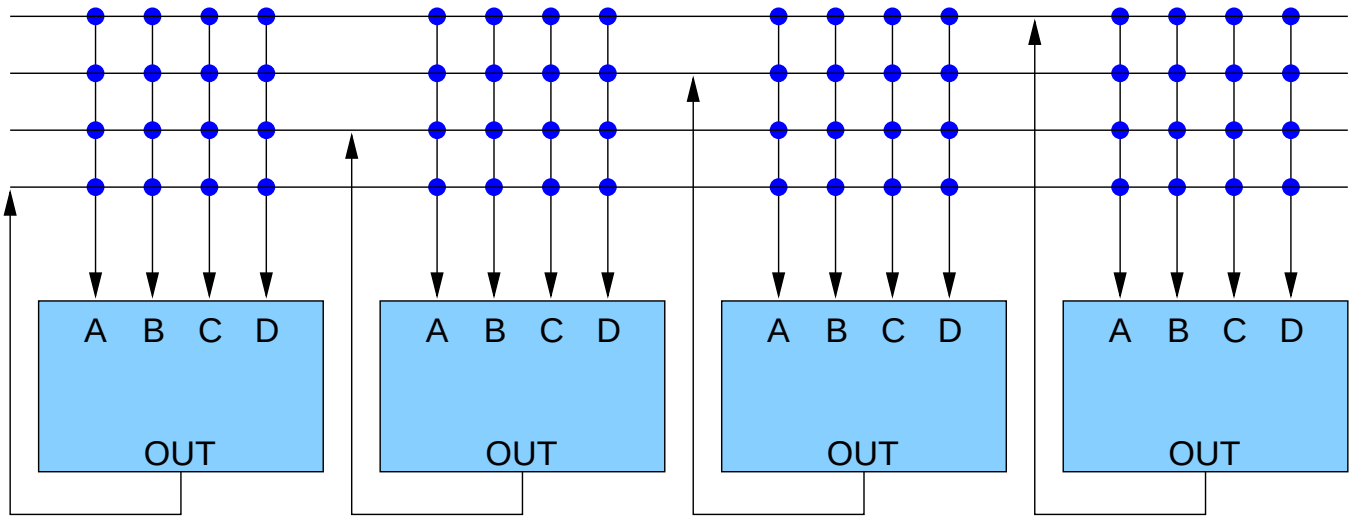
October 2, 1998

A Data Path Element



Each data path element can process four data inputs at each step, as selected by the MUX blocks. The LOGIC blocks provide bit-wise Boolean operations, which can be used to negate one of the addends. The SELECT blocks provide an additional stage of multiplexing. The output can also be shifted one bit position left or right.

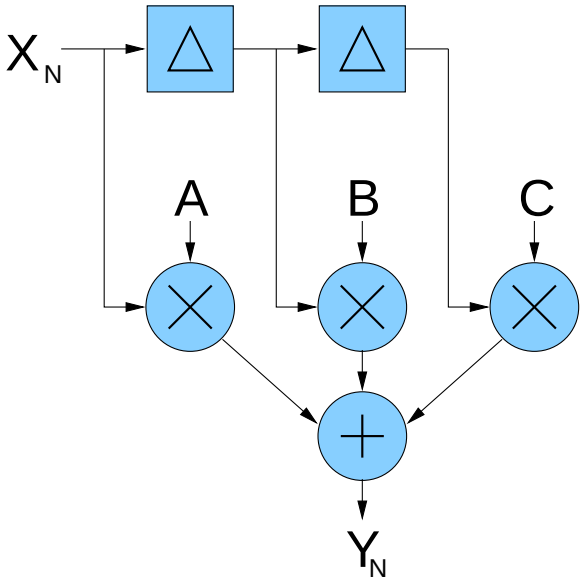
Reconfigurable Data Path Processor



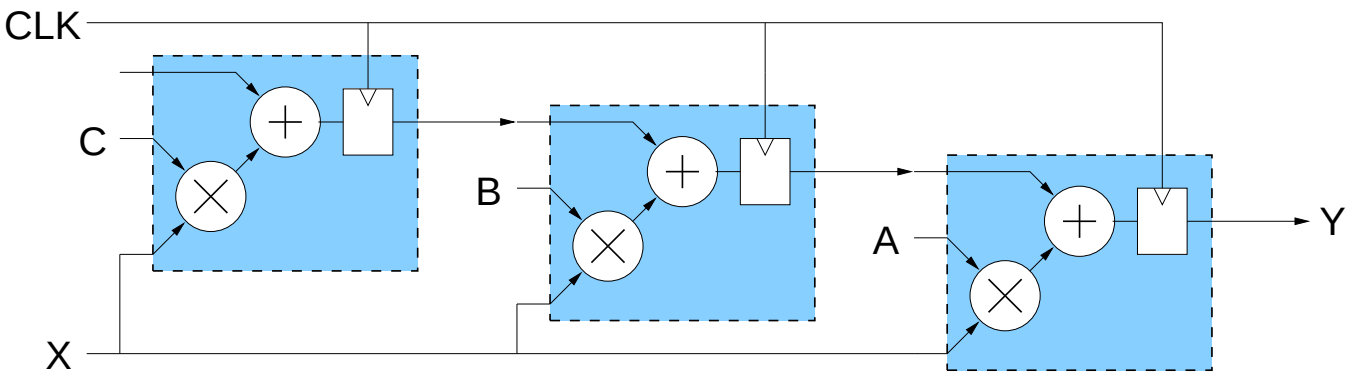
Any number of data path elements can be connected in this fashion. For N processing elements with an M-bit data word the width of the global data bus is $N \times M$. This arrangement allows the data path elements to be connected in any arbitrary fashion.

A Finite Impulse Response Filter

$$Y_N = A \times X_N + B \times X_{N-1} + C \times X_{N-2}$$

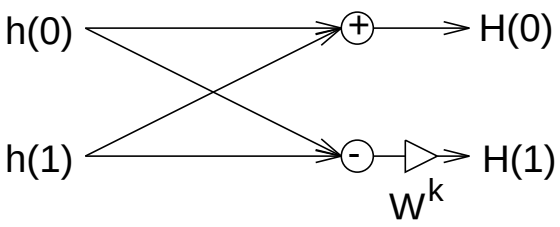


Data Flow Diagram

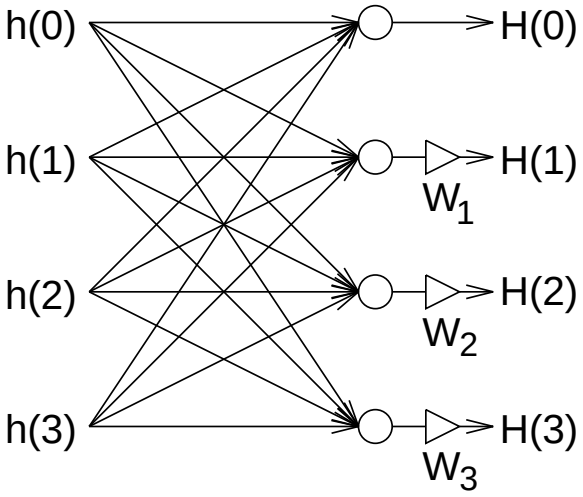


Reconfigurable Data Path Implementation

The Fast Fourier Transform Butterfly



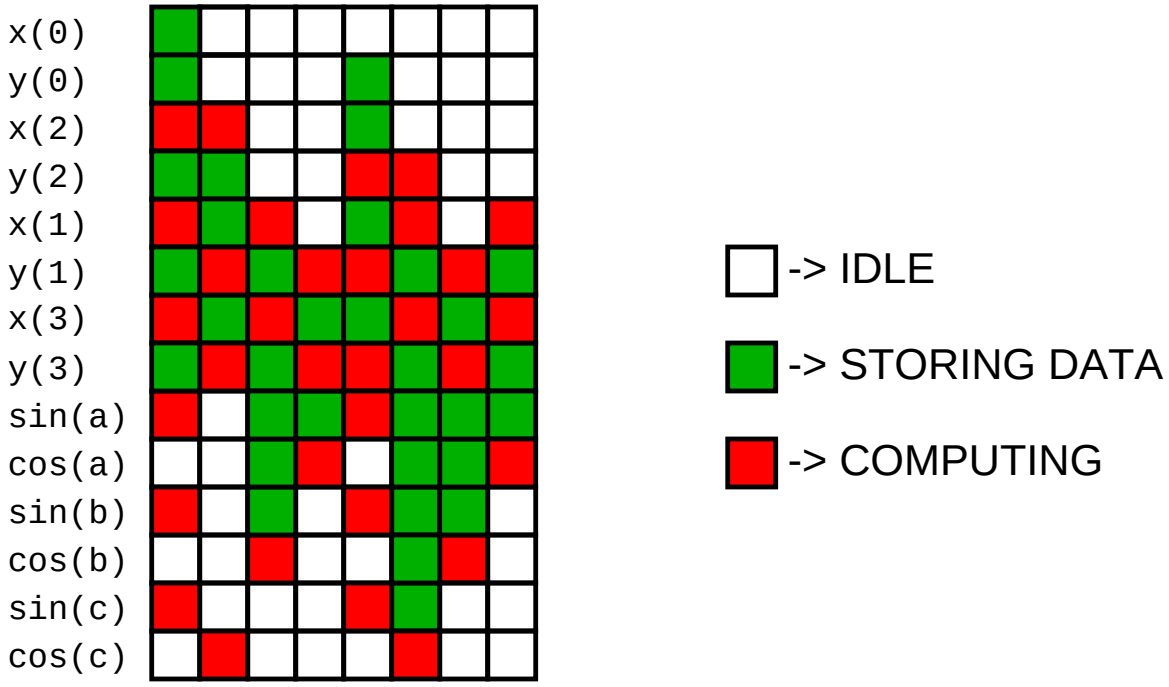
(a) Radix-2



(b) Radix-4

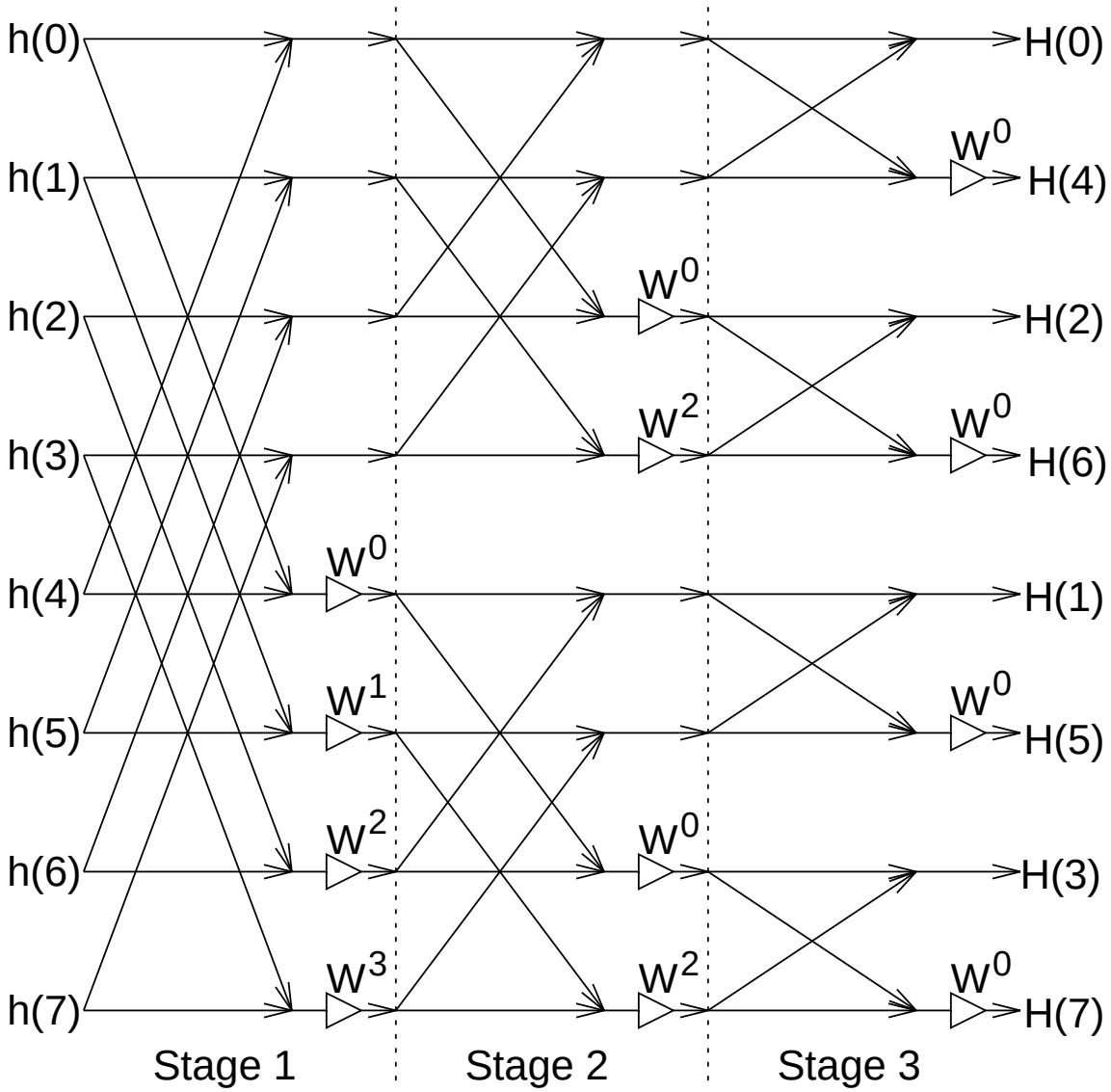
- Fourier Transform maps signals from the time domain to the frequency domain
- Fast Fourier Transform minimizes computation
- Higher radix butterfly reduces total multiplications
- In these examples all data values are complex numbers
- Multiplier is complex

Radix-4 Butterfly on the Reconfigurable Data Path Processor




```
 1     IN;
 2   HOLD,      NOP,      NOP,      NOP,       IN;
 3    +IN,    P0-IN,      NOP,      NOP,     HOLD;
 4   HOLD,     HOLD,      NOP,      NOP,      +IN,    P4-IN;
 5    +IN,     HOLD,    P0-IN,      NOP,     HOLD,      +IN,      NOP,    P5-IN;
 6   HOLD,      -IN,     HOLD,    P1+IN,      +IN,     HOLD,    P4-IN,     HOLD;
 7    +IN,     HOLD,      -IN,     HOLD,     HOLD,      -IN,     HOLD,      +IN;
 8   HOLD,      +IN,     HOLD,      -IN,      +IN,     HOLD,      -IN,     HOLD;
 9  P7*IN,     HOLD,     HOLD,     HOLD,    P3*IN,     HOLD,     HOLD,     HOLD;
10    NOP,     HOLD,     HOLD, *IN+P0<1,      NOP,     HOLD,     HOLD, *IN-P4<1;
11  P6*IN,     HOLD,     HOLD,      NOP,    P2*IN,     HOLD,     HOLD,      NOP;
12    NOP,     HOLD, *IN+P0<1,      NOP,      NOP,     HOLD, *IN-P4<1,      NOP;
13  P5*IN,     HOLD,      NOP,      NOP,    P1*IN,     HOLD,      NOP,      NOP;
14    NOP, *IN+P0<1,      NOP,      NOP,      NOP, *IN-P4<1,      NOP,      NOP;
```

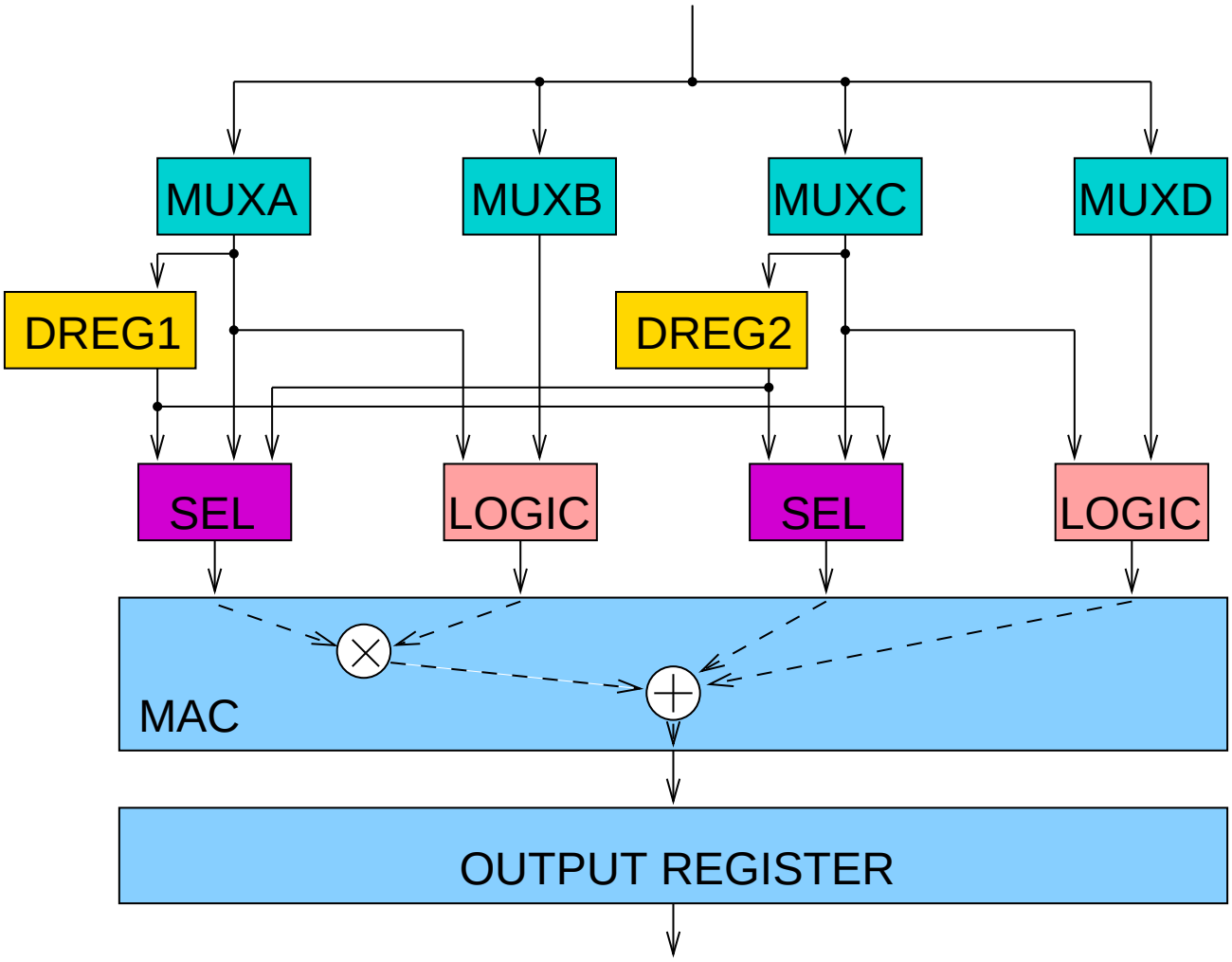
Cascading Butterflies



For an N-point FFT using a radix-M butterfly:

- $\log_M N$ stages are required
- N data reads and writes in each stage

Modified Data Path Element



Modifying the paths from the data registers makes them usable as general data storage. Each data register is available as both a multiplier and an addend.

Performance Comparison

The RDPP requires 63 clock cycles to process 16 complex data points with a cascaded radix-4 algorithm.

Processor	Clock Speed	Clock Cycles	Time, μs
Pentium	133 MHz	4007290	30130
SHARC	40 MHz	72560	1814
RDPP	75 MHz	48384	645
BDSP9124	50 MHz	12490	250

The performance of the RDPP can be greatly improved:

- Double wide I/O to load and store a complex data point on every clock
- Internal coefficient storage, ROM or RAM

Conclusion

- Transform algorithms, such as the Fast Fourier Transform, present unique challenges for parallel processing
- The Reconfigurable Data Path Processor is capable of high performance for transform algorithms
- Simple architectural modifications can significantly improve performance