

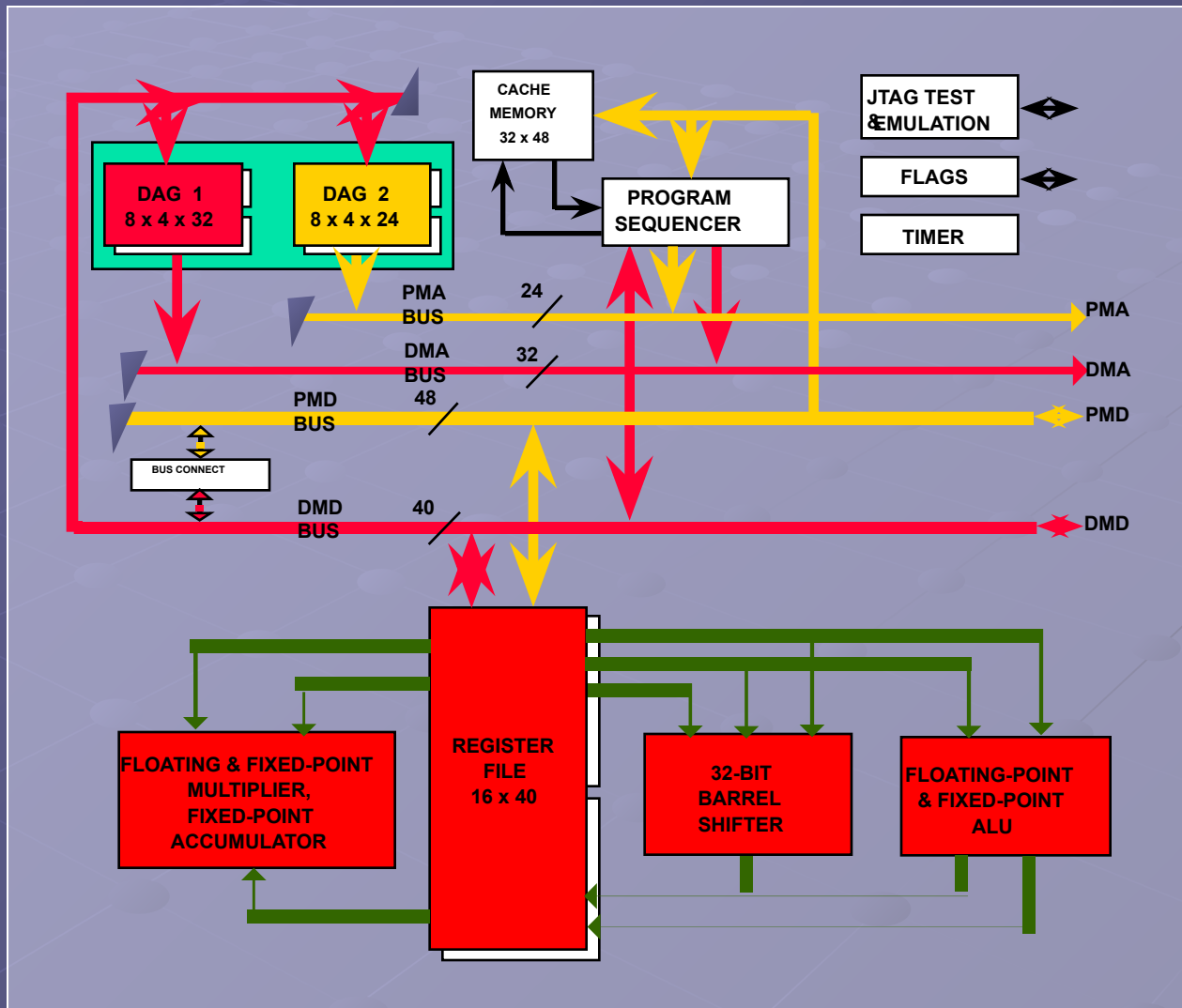
# Blackfin ADSP-21535 Versus Sharc ADSP-21061

By: David W. Rasmussen  
April 15, 2002

# To be covered today:

- Quick overview of the architectures of the both the Blackfin and Sharc DSPs
- Main features of both processors
- Main differences between the processors
- Code sample for an FIR on the Blackfin
- Benchmark comparison of three major DSP algorithms

# Sharc ADSP-21061<sup>[1]</sup>



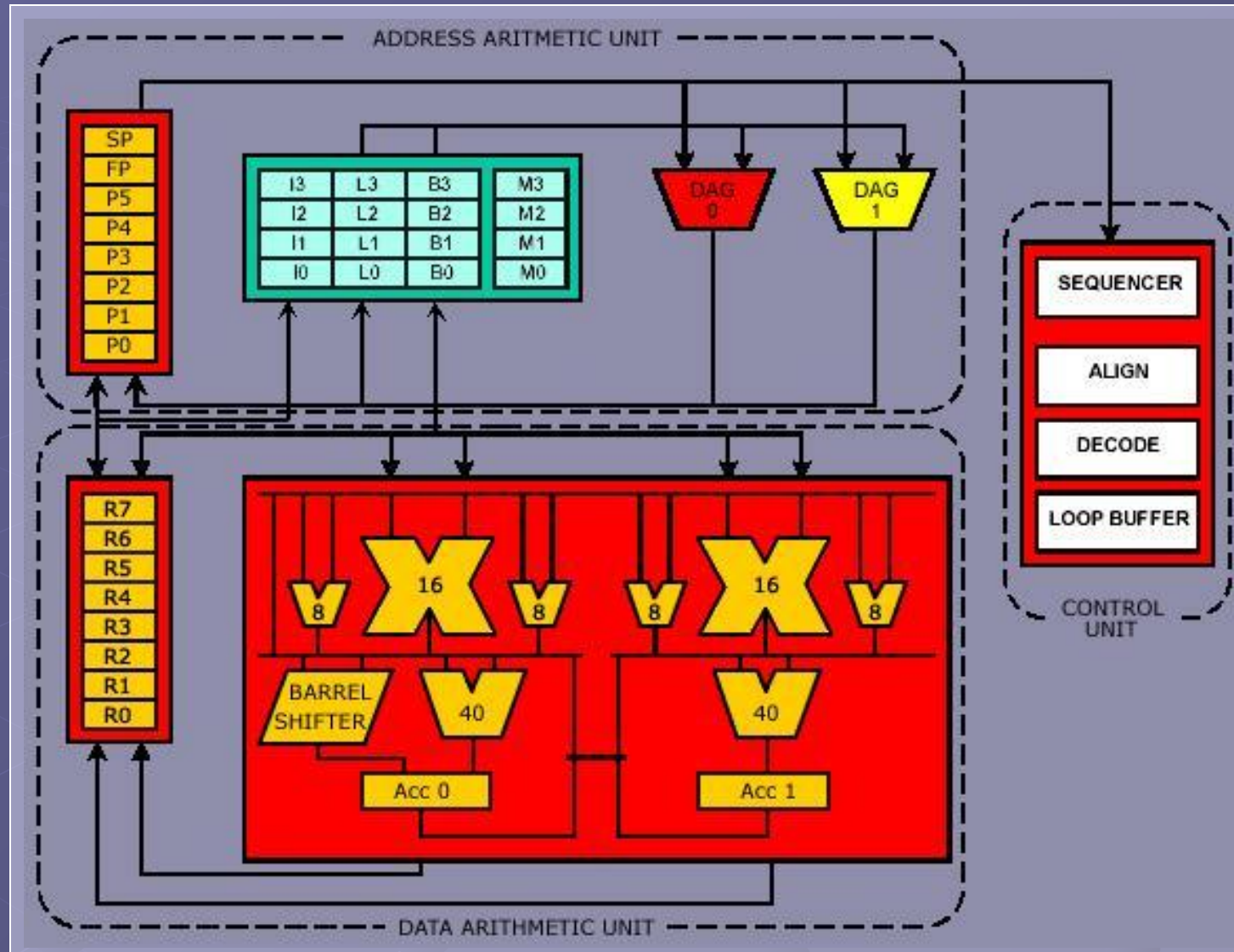
# Sharc's Main Features<sup>[2]</sup>:

- 32/40-bit IEEE floating-point math
- 32-bit fixed-point MACs with 64-bit product and 80-bit accumulation
- No arithmetic pipeline; Thus all computations are single-cycle
- Circular Buffer Addressing supported in hardware
- 32 address pointers support 32 circular buffers
- 16 48-bit Data Registers

# Sharc's Main Features Cont.:

- Six nested levels of zero-overhead looping in hardware
- Four busses to memory (2 DM + 2 PM)
- 1 Mbit on-chip Dual Ported SRAM
- Maximum processing of 50 MIPS
- Possibility of four parallel operations processed in one clock cycle
  - +/-, \*, DM, PM
  - Assuming Pipeline is full
  - PM clashing – utilize Instruction Cache

# Blackfin ADSP-21535<sup>[3]</sup>





# Blackfin's Main Features<sup>[4]</sup>:

- Two 16-bit MACs, two 40-bit ALUs, and four 8-bit Video ALUs
- Support for 8/16/32-bit integer and 16/32-bit fractional data types
- Concurrent fetch of one instruction and two unique data elements
- Two loop counters that allow for nested zero-overhead looping
- Two DAG units with circular and bit-reversed addressing
- 600 MHz core clock performing 600 MMACs

# Blackfin's Main Features Cont.:

- Possibility of the following parallel operations processed in one clock cycle
  - Execution of a single instruction operating on both MACs or ALUs and
  - Execution of two 32-bit Data Moves (either 2 Reads or 1 Read/1 Write) and
  - Execution of two pointer updates and
  - Execution of hardware loop update



# Main Differences:

- The Blackfin is only a 16-bit integer processor, however can operate on 32-bit data values. If 32-bit data value used:
  - Either one or two ALU operations can be performed in one clock cycle
  - One MAC can be obtained however will take more than one clock cycle
- The Sharc is a 32-bit Floating Point processor

# Main Differences Cont.:

- The Blackfin has 4 address registers (with corresponding base, length, and modify) to use for circular buffers versus the Sharc's 32
- The Blackfin has 2 nested hardware loops where the Shark has 6
- The Blackfin has an 8 stage pipeline (fetch 1-2, decode, execute 1-3, writeback) where the Shark has a 3 stage
- The Blackfin is clocked six times faster (300 MHz versus 50 MHz)

# Blackfin FIR Code Sample<sup>[5]</sup>:

```
LSETUP(E_FIR_START,E_FIR_END) LC0=P1>>1; //Loop 1 to Ni/2
E_FIR_START:
    R1=PACK(R1.H,R0.H) || [I0++]=R0 || R2.L=W[I2++];
    //Store X1 into the lower half of R1.
    //Update the delay line.
    //Fetch h0 into lower half of R2
    LSETUP(E_MAC_ST,E_MAC_END)LC1=P2>>1; //Loop 1 to Nc/2 - 1
    A1=R2.L*R1.L, A0=R2.H*R1.H || R2.H=W[I2++] || [I3++]=R3;
    //A1=h0*X1, A0=hn-1*X-n+1.
    //Fetch h1 into upper half of R2.
    //Store the output.
E_MAC_ST:
    A1+=R0.L*R2.H,A0+=R0.L*R2.L || R2.L=W[I2++] || R0=[I1--];
    //A1+=X0*h1, A0+=X0*h0
    //Fetch filter coeff. h2 into the lower
    //half of R2. Fetch X-1 and X-2 into the
    //upper and lower half of R0 (for the
    //first time in this loop)
E_MAC_END:
    A1+=R0.H*R2.L,A0+=R0.H*R2.H || R2.H=W[I2++] ;
    //A1+=X-1*h2, A0+=X-1*h1
    //Fetch h3 into the upper half of R2.
    //(for the first time in this loop)
E_FIR_END:
    R3.H=(A1+=R0.L*R2.H),R3.L=(A0+=R0.L*R2.L) || R0=[P0++] || R1=[I0];
    //A1+=X-n+2*hn-1, A0+=X-n+2*hn+2
    //Fetch the next pair of inputs (X2 and X3) into lower
    //and upper half of R0. Fetch X-n+2 and X-n+3 into R1
```

...

# Benchmarks:

For the Sharc<sup>[6]</sup>

Algorithm Type	Time	Cycles
1024-pt complex FFT	0.37 ms	18,221
FIR Filter (per Tap)	20 ns	1
IIR Filter (per Biquad)	80 ns	4

For the Blackfin<sup>[7]</sup>

Algorithm Type	Time	Cycles
256-pt Complex FFT	0.0106 ms	3,176
FIR Filter (per Tap)	13.33 ns	4
IIR Filter (per Biquad)	20 ns	6

# Analysis:

- Blackfin is faster for the three algorithms
- Unsure of exact performance gain on the FFT (as different lengths) but is somewhere between 2-9 times faster
- Both the FIR and IIR took more cycles to complete on the Blackfin as more cycles are required for 32-bit operations

# References

1. ENCM515 Lecture Slides for January 11, 2002,  
[[http://www.enel.ucalgary.ca/People/Smith/2002webs/encm515\\_02/02presentations/02january/02overviewSHARCarchitecture.ppt](http://www.enel.ucalgary.ca/People/Smith/2002webs/encm515_02/02presentations/02january/02overviewSHARCarchitecture.ppt)], Dr. Mike Smith
2. Sharc Architecture Overview,  
[<http://www.analog.com/technology/dsp/Sharc/architecture.html>], Analog Devices
3. DSP Manuals, [<http://www.analog.com/library/dspManuals/pdf/21535/overview.pdf>], Analog Devices
4. Blackfin Architecture Overview,  
[<http://www.analog.com/technology/dsp/Blackfin/architecture/basics.html>], Analog Devices
5. FIR Blackfin Code Example,  
[[ftp://ftp.analog.com/pub/dsp/blackfin/examples/fir\\_032101.zip](ftp://ftp.analog.com/pub/dsp/blackfin/examples/fir_032101.zip)], Analog Devices
6. Sharc DSP Data Sheet,  
[[http://www.analog.com/productSelection/pdf/ADSP-20161\\_L\\_b.pdf](http://www.analog.com/productSelection/pdf/ADSP-20161_L_b.pdf)], Analog Devices
7. Blackfin DSP Benchmark Comparison,  
[<http://www.analog.com/technology/dsp/Blackfin/benchmarks/examples.html>], Analog Devices



# Special Thanks To:

- Mike Roest for the use of his individual assignment – entitled “Examination of the Analog Devices Blackfin and SHARC 21061”, Submitted March 12, 2002 – as preliminary research material for this report.