LiquidMotion Product Brief

Summary

The LiquidMotion processor is a reconfigurable ASIP (Application Specific Instruction Set Processor) designed to execute user-defined block-matching motion estimation algorithms optimized for hybrid video codecs such as MPEG-2, MPEG-4, H.264 AVC and Microsoft VC-1. The core offers scalable performance dependent on the features of the chosen algorithm and the number and type of execution units implemented. The ability to program the search algorithm to be used, and to reconfigure the underlying hardware that it will execute on, combines to give an extremely flexible motion estimation processing platform.

A base configuration consisting of a single 64-bit integer pipeline, capable of processing 1080p HD video at 30 frames per second using a hexagonal motion estimation search followed by a square refine (as used the x264 [1]) with 1 reference frame and 16x16 block size can be implemented in 2,300 FPGA logic cells. In contrast, a complex configuration including support for motion vector candidates, sub-blocks, motion vector costing using Lagrangian optimization, four integer-pel execution units and one fractional-pel execution unit plus interpolation will need around 14,000 logic cells. A simplified diagram illustrating these two configurations is shown in Fig. 1. At least one integer-pel execution unit must always be present to generate a valid processor configuration but the others units are optional, and are configured at synthesis time.

Features

- ♦ Intuitive and easy programming using a c-like syntax of user-defined block matching motion estimation algorithms.
- ♦ Highly configurable architecture enables the designer to optimize the hardware for the selected algorithm.
- ♦ Binary compatibility so that once an algorithm has been compiled it can be executed by any hardware configuration.
- ♦ Support of advance features such as rate distortion optimization using Lagrangian techniques, sub-partitions and fractional pel searches according to the codec standard.
- ♦ Efficient evaluation of multiple user-defined motion vector candidates transparently to the rest of the algorithm.
- ♦ Toolset available to enable the efficient exploration of the large design space and the generation of the RTL configuration file for the hardware processor library.

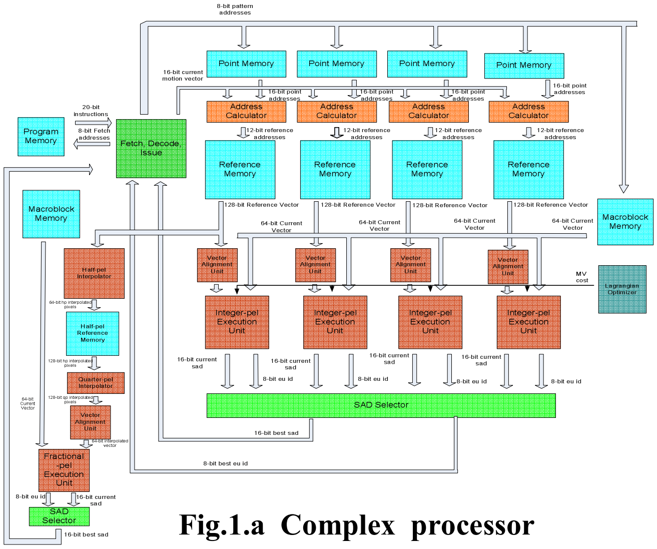


Fig.1.a Complex processor configuration

Applications

- ♦ Video coding (H.264, MPEG-4, MPEG-2, VC-1, AVS)
- ♦ Video enhancement applications such as frame rate conversion, de-interlacing, super-resolution and video stabilization.

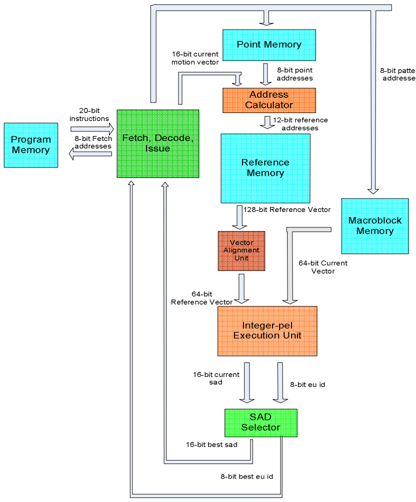


Fig.1.b base processor configuration

Tools

A toolset has been developed that enables the algorithm designer access to the hardware features without any knowledge of the processor microarchitecture. The toolset IDE is shown in Fig.2 as a fully integrated environment composed of a compiler, assembler, cycle accurate model and RTL export. The algorithm designer can create a new algorithm for the required application using typical C constructs such as for, while loops and if-else constructs. The compiler automatically recognises the search points that correspond to fractional-pel searches and generates the correct instructions.

Parallelism is extracted by the compiler by coding search patterns composed of a variable number of search points in a single instruction. The hardware analyses the instruction and distributes the load to the available execution units. Using the cycle accurate model the designer can quickly explore the performance of many configurations in terms of frame per second throughput, compressed video bit-rate and PSNR, hardware complexity and power/energy consumption.

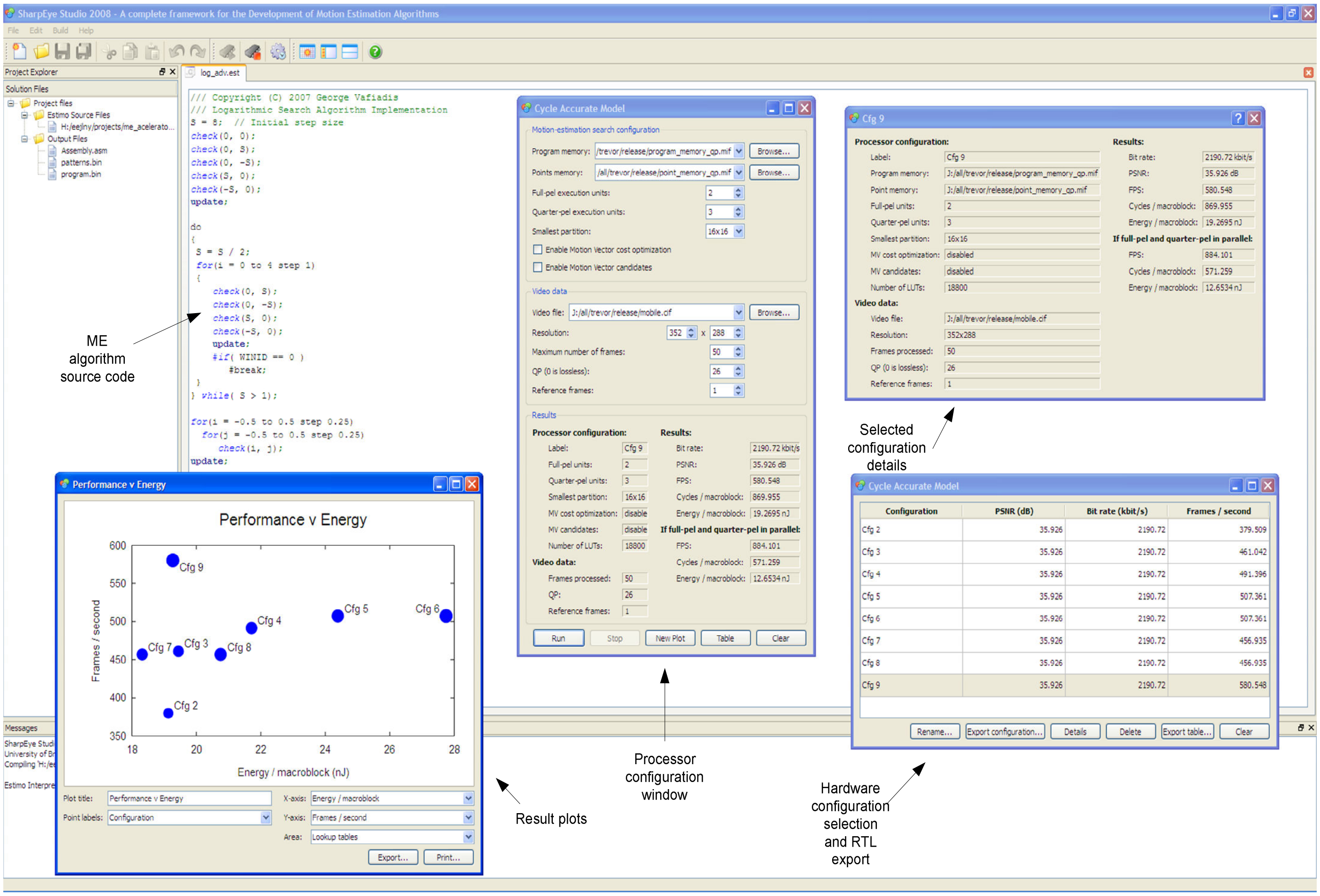
The impact of changes in the original search algorithm can be evaluated before exporting the selected configuration hardware file and program binary. The final implementation can then be generated by processing the configuration file and the rest of the RTL processor description with standard tools such as Synplicity and/or Xilinx ISE.

Deliverables

Toolset (compiler, cycle accurate model and analysis tools), VHDL configurable processor description, VHDL testbench, FPGA prototype implementation using the PCI bus also available together with the original design team.

PROCESSOR CONFIGURATION	CYCLES PER MB (1080P FRAMES PER SECOND)	FPGA COMPLEXITY (LUTS)	MEMORY (BRAMS)
Intel P4 C code (Integer search)	~ 30,000	N/A	N/A
Intel P4 SSE2 assembly (Integer search)	~ 6,000	N/A	N/A
Intel P4 SSE2 assembly (Fractional search)	~ 14,000 (+ ~6000 half pel interpolation)	N/A	N/A
One integer execution unit (Integer search)	785 (30 FPS at 200 MHz)	~2,259	21
One integer and one fractional execution units (Fractional search)	966 (25 FPS at 200 MHz)	~9,143	31

Table 1. Evaluation of fast motion estimation search using 1080P high definition sequences. H.264 integer hexagon search algorithm with square refine, up to 8 MV candidates, Lagrangian MV cost optimization, 112x128 search reference area. Fractional refinement adds 2 diamond half-pel interactions followed by 2 diamond quarter-pel interactions. Intel P4 results included as a computational complexity reference.



```
/// Copyright (C) 2007 George Vafiadis
/// Logarithmic Search Algorithm Implementation
S = 8;  // Initial step size
check(0, 0);
check(0, S);
check(0, -S);
check(S, 0);
check(-S, 0);
update;

do
{
 S = S / 2;
 for(i = 0 to 4 step 1)
 {
    check(0, S);
    check(0, -S);
    check(S, 0);
    check(-S, 0);
    update;
    #if( WINID == 0 )
       #break;
 }
} while( S > 1);

for(i = -0.5 to 0.5 step 0.25)
  for(j = -0.5 to 0.5 step 0.25)
     check(i, j);
update;
```

Cycle Accurate Model

Motion-estimation search configuration

Program memory: /trevor/release/program_memory_qp.mif
Points memory: /all/trevor/release/point_memory_qp.mif
Full-pel execution units: 2
Quarter-pel execution units: 3
Smallest partition: 16x16
□ Enable Motion Vector cost optimization
□ Enable Motion Vector candidates

Video data

Video file: J:/all/trevor/release/mobile.cif
Resolution: 352 x 288
Maximum number of frames: 50
QP (0 is lossless): 26
Reference frames: 1

Results

Processor configuration:
Label: Cfg 9
Full-pel units: 2
Quarter-pel units: 3
Smallest partition: 16x16
MV cost optimization: disable
MV candidates: disable
Number of LUTs: 18800

Video data:
Frames processed: 50
QP: 26
Reference frames: 1

Results:
Bit rate: 2190.72 kbit/s
PSNR: 35.926 dB
FPS: 580.548
Cycles / macroblock: 869.955
Energy / macroblock: 19.2695 nJ

If full-pel and quarter-pel in parallel:
FPS: 884.101
Cycles / macroblock: 571.259
Energy / macroblock: 12.6534 nJ

Cfg 9

Processor configuration:
Label: Cfg 9
Program memory: J:/all/trevor/release/program_memory_qp.mif
Point memory: J:/all/trevor/release/point_memory_qp.mif
Full-pel units: 2
Quarter-pel units: 3
Smallest partition: 16x16
MV cost optimization: disabled
MV candidates: disabled
Number of LUTs: 18800

Video data:
Video file: J:/all/trevor/release/mobile.cif
Resolution: 352x288
Frames processed: 50
QP (0 is lossless): 26
Reference frames: 1

Results:
Bit rate: 2190.72 kbit/s
PSNR: 35.926 dB
FPS: 580.548
Cycles / macroblock: 869.955
Energy / macroblock: 19.2695 nJ

If full-pel and quarter-pel in parallel:
FPS: 884.101
Cycles / macroblock: 571.259
Energy / macroblock: 12.6534 nJ

Cycle Accurate Model

Configuration	PSNR (dB)	Bit rate (kbit/s)	Frames / second
Cfg 2	35.926	2190.72	379.509
Cfg 3	35.926	2190.72	461.042
Cfg 4	35.926	2190.72	491.396
Cfg 5	35.926	2190.72	507.361
Cfg 6	35.926	2190.72	507.361
Cfg 7	35.926	2190.72	456.935
Cfg 8	35.926	2190.72	456.935
Cfg 9	35.926	2190.72	580.548

Plot title: Performance v Energy
Point labels: Configuration
X-axis: Energy / macroblock
Y-axis: Frames / second
Area: Lookup tables

Fig.2 processor development toolset