

# Create Better, Faster, Cheaper Designs for FPGAs

Mobius simultaneously achieves high productivity and QoR comparable to the best HDL designs.

by Per Ljung  
President  
Codetronix LLC  
[ljung@codetronix.com](mailto:ljung@codetronix.com)

With the advent of ever-larger FPGAs, both software algorithm developers and hardware designers must become more productive. Given the frequent lack of resources (time, money, developers) and often unmanageable complexity, today's complex software systems require an increase in abstraction level.

A new generation of methodologies that describe algorithms at a level higher than RTL are collectively called electronic system level (ESL) design tools. ESL enables higher productivity by letting engineers quickly implement their algorithms in hardware. However, many users assume that this is too good to be true, and that the resulting quality of results (QoR) of ESL circuits must be inferior to the best low-level HDL hand designs.

This trade-off between productivity and QoR is no longer necessary. In this article, I'll show you how rapid development using the ESL tool Mobius results in higher productivity, higher performance, and higher quality. I'll also use as examples the QoR of several FPGA cores implemented with Mobius and compare them to traditional HDL designs.

## Mobius

Mobius is a tiny high-level multi-threaded language and compiler designed for the rapid development of hardware/software embedded systems. Mobius compiles your multi-threaded high-level source into synthesizable HDL, from which Xilinx® ISE™ software can generate efficient hardware structures.

A higher abstraction level and ease of use let Mobius users achieve greater design productivity compared to traditional approaches. At the same time, you are not compromising on QoR, because Mobius-generated circuits match the QoR of the best hand designs in terms of throughput and resources for both compact and single-cycle pipelined implementations.

Developing embedded systems with Mobius is much more like software development using a high-level language than hardware development using low-level HDL. Mobius has a fast transaction-level simulator so that the code/test/debugging cycle is much faster than traditional HDL iterations. The compiler-generated HDL is assembled from a set of lego-like primitives connected using handshaking channels. As a result of handshaking, the circuit is robust and correct by construction. Mobius lets software engineers create efficient and robust hardware/software systems, while hardware engineers become much more productive.

Parallelism is the key to obtaining high performance on FPGAs. Mobius enables both compact sequential circuits; latency-intolerant inelastic-pipelined circuits; and parallel latency-tolerant elastic-pipelined circuits. Using the keywords “seq” and “par,” the Mobius language allows basic blocks to be executed sequentially or in parallel. Parallel threads communicate with message-passing channels, where the keywords “?” and “!” are used to read and write over a channel that blocks communication until both the reader and writer are ready. The Mobius compiler also makes development easier by preventing common parallel programming errors such as illegal parallel read/write or write/write statements.

## DES Encryption

DES is a 64-bit block cipher with 16 rounds. It is commonly used as a benchmark because the algorithm is so well specified. Sequential implementations that loop 16 times are common and lead to a compact implementation. For higher throughput, pipelining is required. The entire compact DES and test bench is about 110 lines of Mobius source, and the single-cycle inelastic-pipelined DES is about 120 lines of Mobius source.

Figure 1 shows the Mobius source code for the main loop of the compact DES implementation. First, the counter `rnd` is initialized to 0 simultaneously as the input passes through a pre-round transform.

```
function des(in key0,text0:int64):int64;
var key:int56;
var left,right:int32;
var rnd:integer;
seq
  rnd,{key,left,right}:=0,preround(key0,text0);
  while rnd<16 do
    rnd,{key,left,right}:= rnd+1,round(rnd,key,left,right);
  des:=postround(left,right)
end;
```

Figure 1 – Fragment of Mobius source code for compact DES encryption

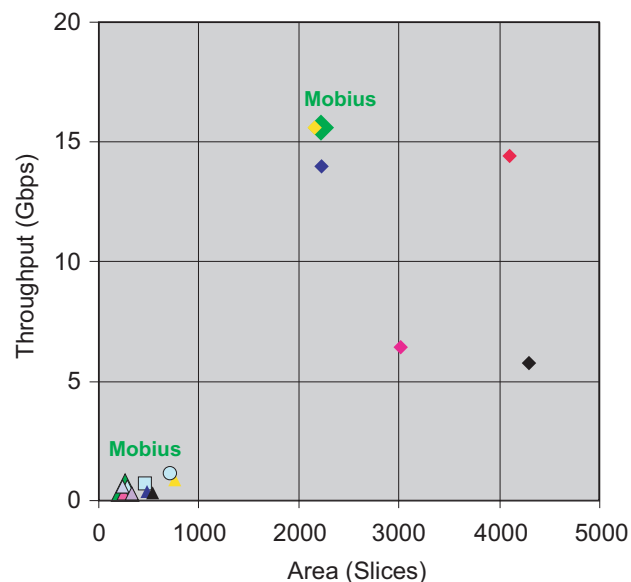


Figure 2 – Resources and throughput of commercial, academic, and open-source DES implementations

Mobius is the only high-level tool that achieves performance similar to HDL hand designs for both compact and inelastic-pipelined implementations.

The synthesizable HDL generated by Mobius lets ISE software infer efficient hardware structures. Table 1 shows the resources and performance for the same Mobius-generated HDL for Virtex-II Pro, Virtex-4, and Virtex-5 targets synthesized using Xilinx ISE v8.2 software. We believe that the 27 Gbps for the Virtex-5 device is the highest DES throughput available for an FPGA.

### AES Encryption

AES-128 is a 128-bit block cipher with 12 rounds that is replacing DES and 3DES. Similar to the DES cipher, AES is an excellent benchmark because the encryption algorithm is completely specified, allowing you to easily compare implementations in terms of resources and throughput. The compact AES-128, including key expansion, is about 220 lines of Mobius source.

Figure 3 shows the area and throughput of several commercial, academic, and open-source AES implementations for a Virtex-II Pro FPGA. All but the Mobius designs are

implemented in HDL, showing that there is a very large QoR variation even between HDL implementations. Note that the QoR variation is substantially greater than the compact DES implementations, which we attribute to HDL designers not being as familiar with the attainable AES performance (it is a newer algorithm). Again, the Mobius QoR are similar to the best HDL hand designs.

### 2D DCT

The 2D discrete cosine transform (DCT) is an example of a DSP filter commonly used in video encoding. Because the 2D DCT is decomposable, it can be expressed as two 1D DCT filters. The Mobius source in Figure 4 effectively demonstrates how you can easily implement streaming algorithms.

Synthesis, placement, and routing show that the resources and throughput of the Mobius 2D DCT design is comparable to the low-level HDL implementation described in Xilinx Application Note XAPP610, "Video Compression Using DCT" ([www.xilinx.com/bvdocs/appnotes/xapp610.pdf](http://www.xilinx.com/bvdocs/appnotes/xapp610.pdf)), as well as other commercial cores.

### Automatic Flow Control

The two 1D DCT and corner-turn memory operate in parallel and use channels to synchronize and communicate data. This elastic pipelining results in automatic flow control. A Mobius handshaking channel comprises probe, request, acknowledge, and data signals to implement automatic flow control between the data producer and consumer. For example, the second `dct1d` starts reading data as soon as the corner-turn memory generates output data. Similarly, the corner turn will read data as soon as the first `dct1d` generates data. The ModelSim handshaking waveforms in Figure 5 show that the first output data is emitted 119 cycles after the first input data.

The throughput of elastic pipelining is slightly lower than inelastic pipelining, but it enables the easy assembly of subsystems with varying latency (including feedback paths), as demonstrated by the conciseness of the 2D DCT source.

	Resources	Clock	Throughput
Virtex-II Pro FPGAs (130 nm)	2,306 Slices 2,122 FFs 3,423 LUTs	261 MHz (Synth) 250 MHz (PAR)	16 Gbps
Virtex-4 FPGAs (90 nm)	3,249 Slices 2,122 FFs 5,413 LUTs	288 MHz (Synth) 281 MHz (PAR)	18 Gbps
Virtex-5 FPGAs (65 nm)	2,122 Slice Regs 1,826 Slice LUTs 2,267 LUT-Flop Pairs	318 MHz (Synth) 417 MHz (PAR)	27 Gbps

Table 1 – Resources and performance of pipelined DES for various targets

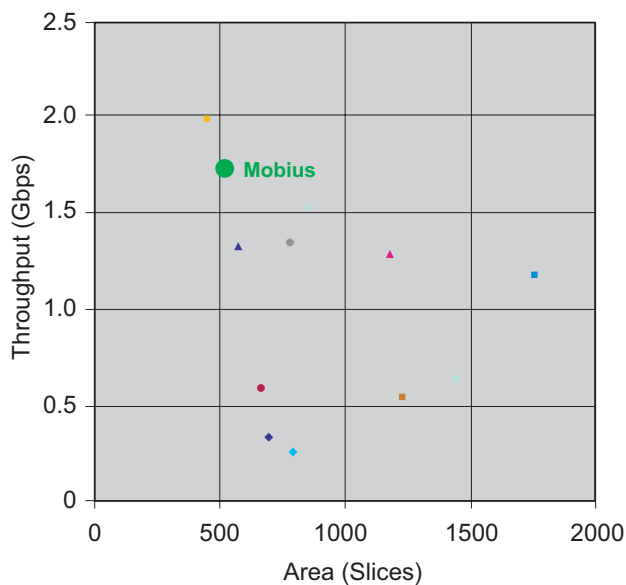


Figure 3 – Resources and throughput of commercial, academic, and open-source AES implementations

Handshaking allows you to ignore low-level timing, since handshaking explicitly determines the control and dataflow. However, it is very easy to understand the timing model of Mobius-generated HDL. Every Mobius signal has a request, acknowledge, and data component where the req/ack bits are used for handshaking. For a scalar variable instantiated as a flip-flop, a read takes zero cycles and a scalar write takes one cycle. An assignment statement takes as many cycles as the sum of its right-hand-side (RHS) and left-hand-side (LHS) expressions. An assignment with scalar RHS and LHS expressions therefore takes one cycle to execute. Channel communications take an unknown number of cycles (waiting until both reader and writer are ready). A sequential block of statements takes as many cycles as the sum of its children, and a parallel block of statements takes the maximum number of cycles of its children.

#### Parameterized Fixed Point and Floating Point

The bit size of the input data, cosine coefficients, intermediate results, and output

data is specified as an 8-bit signed fixed point (with 1 bit for the whole number and 7 bits for the fraction). The definition for the “t” type used in the DCT Mobius design is written as:

```
type t=sfixed(4,8)
```

in the Mobius source. If desired, you can introduce other types, such as specifying that the cosine coefficients use `type sfixed(4,10)`. Mobius also fully supports parameterized floating point.

#### Conclusion

Rapid development using the high-level ESL tool Mobius results in QoR that are comparable to the best low-level HDL hand designs. Mobius users simultaneously benefit from significantly higher design productivity, letting them explore alternative micro-architectures and achieve additional performance goals, resulting in a superior customized circuit.

Using Mobius allows you to rapidly develop high-quality solutions. For more information about using Mobius in your next design, visit [www.codetronix.com](http://www.codetronix.com).

```
procedure dct2d(in a:chan of t; out d:chan of t);
var b,c:chan of t;
par
  dct1d(a,b);
  cornerturn(b,c);
  dct1d(c,d)
end;
```

Figure 4 – Fragment of Mobius source code for 2D DCT

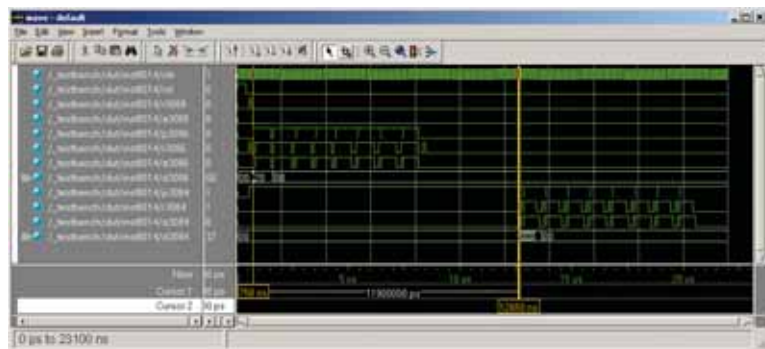


Figure 5 – ModelSim waveforms showing automatic flow control of Mobius 2D DCT

## INTRODUCING



### EMBEDDED MAGAZINE GOES DIGITAL.

With this issue of *Embedded Magazine* we are introducing a new digital edition, which requires no software downloads and is visible on any standard Internet browser.

An enhanced viewing technology lets you browse, search, and make notes, as well as e-mail authors and click through to our advertisers' highlighted websites!

The new digital edition of *Embedded Magazine* combines the award-winning look of our printed magazines with all of the multimedia capabilities the Web has to offer.

We invite you to sign up for your free subscription today!

[www.xilinx.com/xcell/](http://www.xilinx.com/xcell/)

