



# Tiny Tate Bilinear Pairing Core **Specification**

*Author: Homer Hsing*  
*homer.hsing@gmail.com*

**Rev. 0.1**  
**May 3, 2012**

*This page has been intentionally left blank.*

## Revision History

Rev.	Date	Author	Description
0.1	04/28/2012 05/03/2012	Homer Hsing	First Draft Source of citation

# Contents

<b>INTRODUCTION.....</b>	<b>1</b>
<b>ARCHITECTURE.....</b>	<b>3</b>
<b>INTERFACE.....</b>	<b>6</b>
<b>TIMING DIAGRAMS.....</b>	<b>7</b>
<b>FPGA IMPLEMENTATION .....</b>	<b>10</b>
<b>TEST BENCH.....</b>	<b>12</b>
<b>REFERENCES.....</b>	<b>13</b>

## 1

# Introduction

Tiny Tate Bilinear Pairing core is for calculating a special type of Tate bilinear pairing called reduced  $\eta_T$  pairing. Its features are:

- ✓ super-singular elliptic curve  $E: y^2 = x^3 - x + 1$
- ✓ the field is the Galois field  $GF(3^m)$ ,  $m = 97$
- ✓ the irreducible polynomial is  $x^{97} + x^{12} + 2$
- ✓ vendor independent code
- ✓ very low hardware cost ( $\leq 0.2$  US dollar)

## Mathematical background

The elliptic curve  $E: y^2 = x^3 - x + 1$  contains a large cyclic subgroup of prime order  $l$ . Also  $l$  divides  $3^{6m} - 1$  and not any  $3^{j \times m} - 1, j < 6$ , and  $l^2$  does not divide  $\#E$ .<sup>[2]</sup> Now  $E(GF(3^m))$  contains an  $l$ -torsion group  $E[l](GF(3^m))$ , i.e., the set of elements  $P$  of  $E(GF(3^m))$  satisfying  $l \cdot P = O$ , where  $O$  is the point at infinity.<sup>[4]</sup>  $E(GF(3^{6m}))$  also contains an  $l$ -torsion group  $E[l](GF(3^{6m}))$ . The Tate bilinear pairing (of order  $l$ ) is a bilinear map between  $E[l](GF(3^m))$  and  $E[l](GF(3^{6m}))$  to an element of the multiplicative group  $GF(3^{6m})^*$ ,<sup>[2]</sup> i.e.

$$e_l: E[l](GF(3^m)) \times E[l](GF(3^{6m})) \rightarrow GF(3^{6m})^* \quad (1)$$

Let  $P \in E[l](GF(3^m)), Q \in E[l](GF(3^{6m}))$ , let  $f_{l,P}$  be a rational function on the curve with divisor  $l[P] - l[O]$ , there exists a divisor  $D_Q$  equivalent to  $[Q] - [O]$ , with a support disjoint from the support of  $f_{l,P}$ .<sup>[4]</sup> Then the Tate pairing of order  $l$  is defined by

$$e(P, Q) = f_{l,P}(D_Q)^{(3^{6m}-1)/l} \quad (2)$$

It is necessary to raise the value on the right hand side to the power  $\epsilon = (3^{6m} - 1) / l$  to obtain a unique value in  $GF(3^{6m})$ . Consider  $P = (x_1, y_1), Q = (x_2, y_2) \in E[l](GF(3^m))$  i.e.  $x_1, y_1, x_2, y_2 \in GF(3^m)$  and define a distortion map  $\phi$  as

$$\phi(Q) = \phi(x_2, y_2) = (\rho - x_2, \sigma y_2) \quad (3)$$

, where  $\rho, \sigma \in GF(3^{6m})$  satisfying  $\rho^3 - \rho - 1 = 0$  and  $\sigma^2 + 1 = 0$ . Then one can define the modified Tate pairing  $\hat{e}$  for all points  $P, Q \in E[l](GF(3^m))$  as

$$\hat{e}(P, Q) = e_l(P, \phi(Q))^\epsilon = \tau \in GF(3^{6m})^* \quad (4)$$

Generally speaking, the modified Tate pairing is a transformation that takes in two elements in the elliptic curve point group and outputs a nonzero element in the extension field  $GF(3^{6m})$ .

The modified Tate pairing performs in two stages:<sup>[2]</sup>

Stage 1: calculation of  $\phi(Q) \in E[l](GF(3^{6m}))$  in the rational function  $f_P$  on  $E(GF(3^m))$  such that  $div(f_P) \sim l[P] - l[O]$ , i.e.  $e_l(P, \phi(Q)) = f_P(\phi(Q)) = t \in GF(3^{6m})^*$ . This is by Algorithm 1 (Modified Duursma-Lee algorithm) below.

---

Algorithm 1 (Modified Duursma-Lee algorithm)

---

Input:  $P = (x_1, y_1), Q = (x_2, y_2) \in E[l](GF(3^m))$

Output:  $t = e_l(P, \phi(Q)) \in GF(3^{6m})^*$

---

1:  $t = 1 \in GF(3^{6m})^*, \alpha = x_1, \beta = y_1, x = x_2^3, y = y_2^3, \mu = 0 \in GF(3^m), d = m \bmod 3$

2: for  $i$  from 0 to  $m - 1$  do

3:  $\alpha = \alpha^9, \beta = \beta^9, \mu = \alpha + x + d$

4:  $\gamma = -\mu^2 - \beta y \sigma - \mu \rho - \rho^2, t = t^3 \gamma, y = -y, d = d - 1 \bmod 3$

5: next  $i$

6: return  $t$

---

Stage 2: raising  $t$  to the power  $\epsilon_1 = \epsilon/3^{3m} = 3^{3m} - 1$

In [3], Barreto et al. introduced the  $\eta_T$  pairing, which improved above algorithm. The  $\eta_T$  pairing of two points  $P, Q \in E[l](GF(3^m))$  is

$$\eta_T(P, Q) = \begin{cases} f_{T,P}(\phi(Q)), & \text{if } \mu b = -1 \\ f_{-T,-P}(\phi(Q)), & \text{if } \mu b = 1 \end{cases}$$

, where  $b = 1, T = 3^m - \#E(GF(3^m))$  and  $\mu = 1$  if  $m \equiv \pm 1 \pmod{12}$ ,  $\mu = -1$  if  $m \equiv \pm 5 \pmod{12}$ . To assure the obtained value is unique, we have to compute the reduced  $\eta_T$  pairing defined as  $\eta_T(P, Q)^M$  where  $M = (3^{6m} - 1) / \#E(GF(3^m))$ .<sup>[4]</sup>

# 2

## Architecture

### Architecture of the core

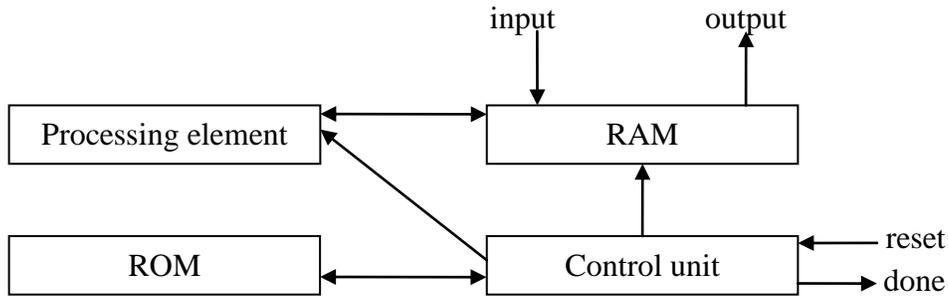


Figure 1: architecture of the core

### Processing element

This is a unified module for addition, subtraction, multiplication and cubing. It has four registers  $R_0, R_1, R_2$  and  $R_3$  supporting following functionalities.

- ✓  $R_3 \leftarrow R_1 + R_2$
- ✓  $R_3 \leftarrow R_1 - R_2$
- ✓  $R_3 \leftarrow (R_1)^{3^{times}}$ , where *times* is specified by instructions.
- ✓  $R_3 \leftarrow R_0 \cdot R_1$

The functionality depends on a control word. The control word  $\{c_0, c_1, \dots, c_{10}\}$  for the processing module is as follows.

Table 1: the control word for the processing element

	$c_0$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$	$c_{10}$
set $R_0$					1	0					

set $R_1$	1	1									
set $R_2$			1	1							
add/sub							1	0	0	0	1
cubic	0	1	0	1	0	0	0	0	0	0	1
mult	0	0	0	0	0	1	1	1	1	1	1

## RAM

This is a dual-port RAM storing 64 words which is 198 bits wide.

## ROM (instruction memory)

This module stores instructions. Each instruction contains five fields:

- ✓ SRC1 (6 bits): address for the first operand
- ✓ SRC2 (6 bits): address for the second operand if available
- ✓ OP (2 bits): operation type
- ✓ TIMES (6 bits): how many times an operation should be repeated
- ✓ DEST (6 bits): address for the operation result

## Control unit

The control unit contains a finite state machine. The states are as follows.

**Table 2: the state of the control unit**

<i>State</i>	<i>Description</i>	<i>successor</i>
START	the default after reset	READ_SRC1
READ_SRC1 ( $S_1$ )	for reading the first operator	READ_SRC2
READ_SRC2 ( $S_2$ )	for reading the second operator	DON or CALC
CALC	for arithmetic	WAIT
WAIT	arithmetic result $\rightarrow R_3$	WRITE
WRITE	$R_3 \rightarrow$ RAM	READ_SRC1
DON	the default after the computation	DON

Its output signals are as follows.

**Table 3: output signals of the control unit**

<i>arithmetic type</i>	<i>state</i>	<i>ram_a_addr</i>	<i>ram_b_addr</i>	<i>pe_ctrl</i>
add/sub	$S_1$	src1	op_add/op_sub	set R0, R1

---

cubic	$S_2$	src2		set R2
	$S_1$	src1	op_cubic	set R0,R1,R2
mult	$S_2$	src2		
	$S_1$	src1		set R1,R2
	$S_2$	src2	src2	set R0

---

# 3

## Interface

The Tiny Tate Bilinear Pairing core implements the signals shown in the table below.

Input signals are synchronous and sampled at the rising edge of the clock.

Output signals are driven by flip-flops, and not directly connected to input signals by combinational logic.

For signals wider than 1 bit, the range is most significant bit down to least significant bit.

**Table 4: Interface signals**

<i>Signal name</i>	<i>Width</i>	<i>In/Out</i>	<i>Description</i>
clk	1	In	clock
reset	1	In	<i>active high synchronous</i> reset
sel	1	In	active high for reading/writing RAM from outside
addr	6	In	RAM address
w	1	In	active high for writing RAM; low for reading RAM
update	1	In	active high for updating the I/O buffer
ready	1	In	active high if it is ready to read/write
i	1	In	serial input
o	1	Out	serial output whose value is valid when the <i>done</i> signal is asserted
done	1	Out	The termination signal. It is inactive low after the <i>reset</i> signal asserted, and is active high only if the Tate pairing computation is done.

## 4

# Timing

The entire computing process is

- ✓ write input to RAM
- ✓ wait for the end of the calculation, the time when the *done* signal is asserted
- ✓ read the result from RAM

## Memory address of the input/output

Assume that the input is  $P = (x_P, y_P)$ ,  $Q = (x_Q, y_Q)$ , assume the output is  $\eta_T(P, Q)^M = T$ ,  $T = t_0 + t_1\sigma + t_2\rho + t_3\rho\sigma + t_4\rho^2 + t_5\rho^2\sigma$ , assume each of  $\{x_P, y_P, x_Q, y_Q, t_0, t_1, \dots, t_5\}$  is 198-bit wide, equal to the width of RAM. The memory address for input/output is defined below.

**Table 5: memory address for input/output**

<i>input/output</i>	<i>address</i>
$x_P$	3
$y_P$	5
$x_Q$	6
$y_Q$	7
$t_0$	9
$t_1$	10
$t_2$	11
$t_3$	12
$t_4$	13
$t_5$	14

## Write to RAM

In order to write a value  $V[197:0]$  to a specified RAM address  $Addr[5:0]$ , firstly let  $sel = 1, addr = Addr[5:0], w = 1$ , and comply with the timing below.

**Table 6: timing for writing to RAM**

<i>clock cycle</i>	<i>signal name/value</i>		
	update	ready	i
$T$	1	0 or 1	0 or 1
$T + 1$	0	1	$V[0]$
$T + 2$	0	1	$V[1]$
	.....		
$T + 198$	0	1	$V[197]$

Then let  $w = 0$ .

When write the RAM, the *reset* signal should be asserted and keep asserted until the termination of writing.

## Wait for the end of the computation

Wait until the signal *done* is active.

## Read from RAM

In order to read from a specified RAM address  $Addr[5:0]$ , firstly let  $sel = 1, addr = Addr[5:0], w = 0$ , and comply with the timing below.

**Table 7: timing for reading from RAM**

<i>clock cycle</i>	<i>signal name/value</i>		
	update	ready	o
$T$	1	0 or 1	0 or 1
$T + 1$	0	1	$V[0]$
$T + 2$	0	1	$V[1]$
	.....		
$T + 198$	0	1	$V[197]$

Then keep  $w = 0$ .

You may use a 198 bits wide register to record  $V$ .

# 5

## FPGA Implementation

### Synthesis results (ISE)

Table 8: Synthesis result (ISE)

<i>Device</i>	<i>Xilinx Spartan 3 XC3S200-5PQ208</i>
Number of Slice Flip Flops	1,319
Number of 4 input LUTs	3,028
Number of occupied Slices	1,730
Number of bonded IOBs	15
Minimum period	10.455ns
Maximum Frequency	95.6MHz

\* Synthesis tool is Xilinx ISE 13.4.

\* Related Xilinx synthesis configuration files are in the directory `synthesis/xilinx`.

### Synthesis results (Quartus)

Table 9: Synthesis result (Quartus)

<i>Device</i>	<i>Altera Cyclone II EP2C20F484C7</i>
Total logic elements	3,637
Dedicated logic registers	1,310
Total memory bits	25,984
Total pins	15

\* Synthesis tool is Altera Quartus II 11.1.

### Speed

The core computes one Tate pairing in 1.05 milliseconds if with a 50MHz clock.

## Hardware price

The core applies to Xilinx Spartan 3 XC3S200. The price of that FPGA <sup>[5]</sup> is less than 0.2 US dollar per piece.

## Compared to Tate Bilinear Pairing core <sup>[6]</sup>

**Table 10: Compared to Tate Bilinear Pairing core**

	<i>Tiny Tate Bilinear Pairing core</i>	<i>Tate Bilinear Pairing core</i>
Device	Xilinx Spartan 3 XC3S200	Xilinx Virtex 4 XC4VLX200
Number of Slice Flip Flops	1,319	31,383
Number of 4 input LUTs	3,028	47,083
Number of occupied Slices	1,730	30,149
Computation time	1.02ms	0.76 ms

# 6

## Test bench

The file “testbench/simulation.do” is a batch file for ModelSim to compile the HDL files, setup the wave file, and begin function simulation. In order to make it work properly, the working directory of ModelSim must be the directory of “testbench”.

The file “testbench/test\_pairing.v” is the main test bench for the Tate Bilinear Pairing core. The test bench is self-checked. It feeds input data to the core and compares the correct result with the output of the core. If the output is wrong, the test bench will display an error message.

If the function of the core is wrong, some other self-checking test benches in the “testbench” directory may help. The object of each test bench is listed in the table below.

**Table 11: the object being tested by each test bench**

<i>File name</i>	<i>the object being tested</i>
test_const.v	the module outputting constant and the control word for the processing element
test_fsm.v	control unit
test_pe.v	processing element
test_ram.v	RAM
test_tiny.v	the module computing the Tate bilinear pairing

# 7

---

## References

- [1] I.Duursma, H.S.Lee. Tate pairing implementation for hyper-elliptic curves  $y^2 = x^p - x + d$ . In Advances in Cryptology Proc. ASIACRYPT'03, pp. 111-123, 2003.
- [2] T.Kerins, W.P.Marnane, E.M.Popovici, and P.S.L.M.Barreto. Efficient hardware for the Tate pairing calculation in characteristic three. In Cryptographic Hardware and Embedded Systems Proc. CHES '05, pp. 412-426, 2005.
- [3] P.Barreto, S.Galbraith, C.O hEigartaigh, and M.Scott. Efficient pairing computation on supersingular abelian varieties. In Designs, Codes and Cryptography. Springer Netherlands, Mar. 2007, vol. 42(3), pp. 239–271.
- [4] J.Beuchat, N.Brisebarre, J.Detrey, E.Okamoto, M.Shirase, and T.Takagi. Algorithms and arithmetic operators for computing the  $\eta_T$  pairing in characteristic three. In IEEE Transactions on Computers, Special Section on Special-Purpose Hardware for Cryptography and Cryptanalysis, 57(11):1454-1468, 2008.
- [5] Price of Xilinx Spartan 3 FPGA. <http://www.alibaba.com/showroom/xc3s200.html>
- [6] Tate Bilinear Pairing core @ opencores.org. <http://opencores.org/project,pairing>