**INTERFACE CONTROL DOCUMENT**

**FOR**

**HARDWARE IMPLEMENTATION OF RSA 4096 Modular Exponentiation**

**APPLIED CRYPTOLAB UCSB**

**DEPARTMENT OF COMPUTER SCIENCE**

**UNIVERSITY OF CALIFORNIA, SANTA BARBARA**

***Version:*** 0.1

***Last Modified:*** Balasubramaniam Muthuvelu(balasubramaniam@umail.ucsb.edu)

***Document Number:*** 1

***Document Approved by:*** Prof. Çetin Kaya KoÇ (koc@cs.ucsb.edu)

|  |  |
| --- | --- |
| College Logo | http://www.aw.id.ucsb.edu/logos/images/ucsbwave_144.jpg |

Contents

[Version Control History 3](#_Toc355035272)

[1. Introduction 4](#_Toc355035273)

[1.1 RSA Algorithm 4](#_Toc355035274)

[1.2 RSA Implementation 4](#_Toc355035275)

[1.3 Montgomery transform based modular exponentiation algorithm 4](#_Toc355035276)

[1.4 MonPro Algorithm 7](#_Toc355035277)

[2. Device Details 10](#_Toc355035278)

[2.1 Memory Operations 10](#_Toc355035280)

[2.1.1 Control Signals 10](#_Toc355035281)

[2.1.2 Byte Enable Support 11](#_Toc355035282)

[2.1.3 Address Clock Enable 11](#_Toc355035283)

[2.1.4 Memory Modes 11](#_Toc355035284)

[2.1.5 Input/Output clock mode 12](#_Toc355035285)

[2.1.6 Read/Write Clock Mode 13](#_Toc355035286)

[2.1.7 Power-Up conditions & memory initialization 13](#_Toc355035287)

[2.1.8 Single Clock Mode 13](#_Toc355035288)

[3. Implementation Summary 14](#_Toc355035289)

[4. Appendix A 15](#_Toc355035290)

[4.1.1 Pin assignment details from initial simulation 15](#_Toc355035294)

[4.1.2 Usage details: 28](#_Toc355035295)

# Version Control History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author of Changes | Approver | Description of Changes |
| 0.1 | 24 April 2013 | Balasubramaniam Muthuvelu | Cetin Kaya Koc | Initial draft Document |
| 0.2 | 1 May 2013 | Qin ZHOU |  | Multi-add, compilation report correction |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction

## RSA Algorithm

The RSA algorithm, invented by Rivest, Shamir, and Adleman, is one of the simplest public-key cryptosystems. The parameters are n, p and q, e, and d. The modulus n is the product of two distinct large random primes p and q such that *n = pq*. The public exponent *e* is a number in the range *1 < e < Ф(n)* such that

*GCD(e,Ф(n)) = 1*

where *Ф(n)* is the Euler's totient function of n, given by

*Ф(n) = (p-1)(q-1)*

The private exponent of *d* is obtained by inverting *e* modulo *Ф(n)* i.e.

*d=e-1 mod Ф(n)*

Extended Euclidean algorithm is one way of finding the value of d. The encryption operation is performed by computing

*C = Me (mod n)*

where *M* is the plaintext such that *0 < M < n*. The number *C* is the ciphertext from which the plaintext *M* can be computed using

*M = Cd (mod n)*

The RSA algorithm can be used to send encrypted messages and to produce digital signatures for electronic documents. It provides a procedure for signing a digital document, and verifying whether the signature is indeed authentic. The signing of a digital document is somewhat different from signing a paper document, where the same signature is being produced for all paper documents. A digital signature cannot be a constant; it is a function of the digital document for which it was produced. After the signature (which is just another piece of digital data) of a digital document is obtained, it is attached to the document for anyone wishing to verify the authenticity of the document and the signature.

## RSA Implementation

There are many optimized software and hardware techniques that are available currently, using which RSA algorithm can be implemented. In our case, we are trying to implement a hardware algorithm which would be optimal with respect to time and which can encrypt and decrypt a given message when the modulus size is 4096 bits. We have implemented the algorithm using a Montgomery-transformed exponentiation algorithm the details of which are provided in the next section.

## Montgomery transform based modular exponentiation algorithm

The algorithm of choice for the hardware implementation is the Montgomery-transformed exponentiation using 128-word integers(s) with a word size(w) of 32 bits and the Montgomery constant r where r = 2sw . The algorithm has been implemented using 1-bit binary exponentiation.

The input variables are as provided below:

1. Input message *m,* signed integer of size 4096 bits. Generally, the messages would be greater than (or less than) 4096 bits in size. In cases where the size is greater than 4096 bits we divide the message into blocks of 4096 bits. If any of the blocks is less than 4096 bits we leave the message as it is. Since, this is signed integer notation, we the most significant bit of message m would be 0 when m>0 due to properties of two’s complement notation.
2. The exponent *e* which is also of size 4096 bits. Exponent *e* is k-bit unsigned binary number for k≤4096. Inputs m and e constitute our primary inputs.
3. The modulus operator *n* which is of size 4096 bits. *n* is a 4096 bit signed binary number. Also, n is assumed to be an odd number such that the least significant bit n0 =1 and the most significant bit of message *n* would be 0 when *n* >0 due to properties of two’s complement notation.
4. The input *n0|* which is pre-computed from *n* and *w* such that

*n0| = -n-1 (mod 2w)*

1. The Montgomery constant *r* defined as

*r = 2sw (mod n)*

1. The constant *t* defined as

*t = r2 (mod n) or t = 22sw (mod n)*

The inputs *n0|* is one word in size (32 bits) while *r* and *t* are signed integers of size 4096 bits. The output of the process is the cipher text *c* of size 4096 bits.

The ‘process’ as mentioned earlier is implemented using a variation of Montgomery multiplication method.

When operating on integers of size 4096 bits, the Montgomery method used here derives the output using the steps in the flowchart described below.

i > 0

e.i = 1

Start

Read

## MonPro Algorithm

The MonPro algorithm takes two inputs: x and y and computes the output z such that each variable holds an s-word (total sw bit) signed integer. In this case x and y are 128 words of 32 bit each. The initial value of z is initialized to zero. Hence,

x = x127x126….x1x0

y = y127y126….y1y0

z = z127z126….z1z0

The secondary inputs *n0|, n, r and t* are also assumed to be available. We have

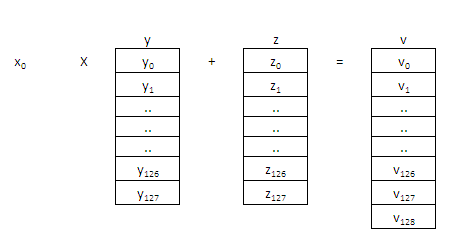
n = n127n126….n1n0

n0| = n0

where n has 128 words of 32 bit each and n0| has 1 word of 32 bit.

The steps of the MonPro Algorithm are as explained below.

*Step 1:* We take the LSW of x, namely x0 and multiply by the 128-word y, and add it to the 128 word partial product z (which is now all zero) to obtain the (129)-word temporary result v as



*Step 1a:* The computation in Step 1 is accomplished using a Multiply-Add block that multiplies two 1-word numbers (x0 and y0), producing a 2-word number and adding another 1-word number (z0) and a higher word (C0) to get (C1, S0); the lower word (S0) is assigned to value (v0) and the higher word (C1) is used for next Multiply-Add step, as follows:

(C1,S0) = x0.y0 + z0 + C0

v0 = S0

(C2,S1) = x0.y1 + z1 + C1

v1 = S1

(C3,S2) = x0.y2 + z2 + C2

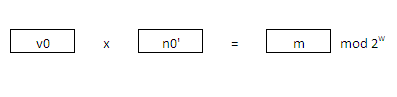
v2 = S2

…

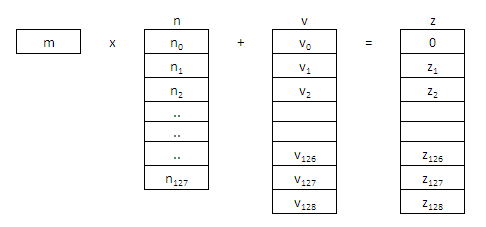
Where the initial value of C0 = 0

*Step 2*: We then take the LSW of v, namely v0 and multiply by the 1-word n0| modulo 2w and obtain the 1-word integer *m* as

m = n0|.v0 (mod 2w)



*Step 3*: We take the 1-word m and multiply by the 128-word n, and add it to the 129 word temporary value v (from step-1) to obtain the new partial product which is the 129 word z.



*Step 3a*: The computation in step 3 is accomplished using a Multiply-Add block that multiplies two 1-word numbers (m and n0), producing a 2-word number and adding another 1-word number (v0) and a higher word (C0) to get (C1, S0); the lower word (S0) is assigned to value (z0) and the higher word (C1) is used for next Multiply-Add step, as follows:

(C1,S0) = m.n0 + v0 + C0

z0 = S0

(C2,S1) = m.n1 + v1 + C1

z1 = S1

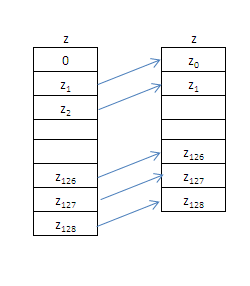
(C3,S2) = m.n2 + v2 + C2

z2 = S2

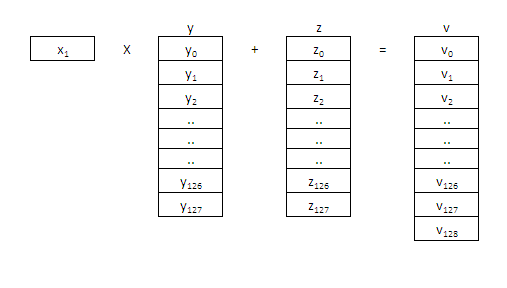
…

Such that the initial value C0 = 0.

*Step 4*: The resulting partial product z has its LSW set to zero, due to the Montgomery property, and therefore, we shift z to obtain the new 128 word partial product.



*Step 5*: In the next step the next word of x, namely x1 is taken and multiplied with the 128 word y and added to the partial product z to obtain the new temporary result v.



*Step 6*: This is followed up by computing the new m value

m = n0|.v0 (mod 2w)

and then the multiplication of m by n, and then the addition of the result to v to obtain the new 129 word partial product z and shifting the value of z by one word (since z0 = 0) to obtain the new 128 word z value.

*Step 7*: Proceeding in this way, we multiply all xi values by the multiplicand y and reduce it modulo n for i = 1,2,3 to 127.

# Device Details

The modular exponentiation routine using Montgomery algorithm has been implemented as a standalone functionality using an FPGA. The device takes primary inputs *m* and *e* of size 4096 bits respectively, secondary inputs of size *n0|,r and t* of size 32 bits,4096 and 4096 bits respectively and produces an output *c* of size 4096 bits. The device which we opted to implement this is Altera Cyclone II EP2C50. Please note that the secondary inputs are generally constants that can be retained for a number of cycles where the primary inputs vary. For instance, given a message of 16k bits in size the RSA encryption would split the message into 4 blocks of 4k each (which is the maximum possible size). For all the cycles involved, though the message block varies, the secondary inputs can be the same and hence need not be re-entered through the inputs pins for each individual message blocks. They can be initialized at the beginning of the encryption process for this message.

Altera Cyclone II EP2C50 has high density architecture with around 50528 logic elements. The various parameters of this device which would of interest to the reader are as follows.

* There are 129 M4K RAM blocks (4Kbits plus 512 parity bits). The device supports various input configurations ranging from 4k \* 1 bit to 128 \* 32 bits at a time.
* There are 86 18-bit \* 18-bit multipliers.
* There are up to 450 user I/O pins including 16 dedicated clock pins that can be used for data inputs.
* The maximum performance of device is at 250 MHz.

The parameters at which we are operating the device are as follows.

Data input width: 128\* 32 bits

Clock Frequency: 100 MHz

The various pin details including pin details for input, output and power considerations (voltage supply and ground) and included in the appendix.



## Memory Operations

The device features embedded memory structures to address on-chip memory needs of FPGA designs. The embedded memory structure consists of columns of M4K memory blocks that can be configured to provide various memory functions.

### Control Signals

The Clock enable signal controls the clock entering the entire memory block, not just the input and output registers. The signal disables the clock so that the memory block does not see any clock edges and will not perform any operations.

There are three ways to reset the registers in the M4K blocks: power up the device, use the *aclr* signal for output register only, or assert the device-wide reset signal using the *DEV\_CLRn* option.

Cyclone II devices do not support asynchronous clear signals to input registers. Only output registers support asynchronous clears. When applied to output registers, the asynchronous clear signal clears the output registers and the effects are seen immediately.

### Byte Enable Support

All M4K memory blocks support byte enables that mask the input data so that only specific bytes of data are written. The unwritten bytes retain the previous written value. The write enable (wren) signals, along with the byte enable (byteena) signals, control the RAM block’s write operations. The default value for the byte enable signals is high (enabled), in which case writing is controlled only by the write enable signals. There is no clear port to the byte enable registers. Since we are using a data width of 32 bits byteena[3:0] has to be asserted as 4’b1111 (default) and write enable has to be asserted high to enable writing.

### Address Clock Enable

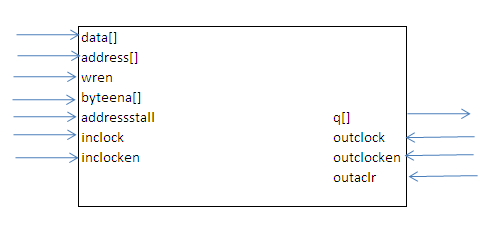
Cyclone II M4K memory blocks support address clock enables, which holds the previous address value until needed. The address register output is fed back to its input via a multiplexer. The multiplexer output is selected by the address clock enable (*addressstall*) signal. Address latching is enabled when the *addressstall* signal goes high (active high). The output of the address register is then continuously fed into the input of the register until the *addressstall* signal goes low. Hence, *addressstall* can be enabled high to hold a previous address value till needed.

### Memory Modes

The M4K supports various memory modes of operation. We estimate that two modes of operations would be significant in our process. These are single port and simple dual port. Single port mode supports non-simultaneous read and write operations. Figure 1 shows the single-port memory configuration for cyclone II memory blocks.

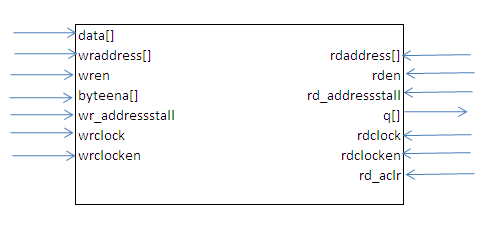
In single-port mode, the outputs are in read-during-write mode, which means that during the write operation, data written to the RAM flows through to the RAM outputs. When the output registers are bypassed, the new data is available on the rising edge of the same clock cycle on which it was written. The device is being operated on a port width configuration of 128 \* 32 bits.

Simple dual port supports simultaneous read and write operation. Figure 2 shows the simple dual port memory configuration



1. Single-Port Mode

Simple dual port mode supports simultaneous read and write operation. Figure 2 shows the simple dual port memory configuration. In simple dual-port mode, the memory blocks have one write enable and one read enable signal. They do not support a clear port on the write enable and read enable registers. When the read enable is deactivated, the current data is retained at the output ports. If the read enable is activated during a write operation with the same address location selected, the simple dual-port RAM output is the old data stored at the memory address. However, the widest bit configuration of M4K blocks in true dual port is 256\*16-bit.



2. Simple Dual Port Mode

### Input/output clock mode

Cyclone II memory blocks can implement the input/output clock mode for true and simple dual-port memory. On each of the two ports, A and B, one clock controls all registers for the data, write enable, and address inputs into the memory block. The other clock controls the blocks’ data output registers. Each memory block port also supports independent clock enables for input and output registers. Asynchronous clear signals for the registers are not supported.

### Read/Write Clock Mode

Cyclone II memory blocks can implement read/write clock mode for simple dual-port memory. The write clock controls the blocks’ data inputs, write address, and write enable signals. The read clock controls the data output, read address, and read enable signals. The memory blocks support independent clock enables for each clock for the read- and write-side registers. This mode does not support asynchronous clear signals for the registers.

### Power-Up conditions & memory initialization

The Cyclone II memory block outputs always power-up to zero, regardless of whether the output registers are used or bypassed. Even if an MIF pre-loads the contents of the memory block, the outputs still powers up cleared. For example, if address 0 is pre-initialized to FF, M4K blocks power up with the output at 00. A subsequent read after power up from address 0 outputs the pre-initialized value of FF.

### Single Clock Mode

Cyclone II memory blocks support single-clock mode for simple dual-port, and single-port memory. In this mode, a single clock, together with a clock enable, controls all registers of the memory block. This mode does not support asynchronous clear signals for the registers.

# Implementation Summary

The primary and secondary input values are assumed to be copied to the register locations within the FPGA. We assume that the values will be copied to the below mentioned locations. The output c will then be available in location as mentioned below.

|  |  |  |  |
| --- | --- | --- | --- |
| variable | class | size(bits) | starting address location |
| m | input | 4096 | 0x0000 |
| e | input | 4096 | 0x1000 |
| n | input | 4096 | 0x2000 |
| r | input | 4096 | 0x3000 |
| t | input | 4096 | 0x4000 |
| n0' | input | 32 | 0x5000 |
| c | input | 4096 | 0x6000 |

# Appendix A



### Pin assignment details from initial simulation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pin Name/Usage | Location | Dir. | I/O Standard | Voltage | I/O Bank | User Assignment |
| GND | A1 | gnd |  |  |  |  |
| VCCIO3 | A2 | power |  | 3.3V | 3 |  |
| GND\* | A3 |  |  |  | 3 |  |
| GND\* | A4 |  |  |  | 3 |  |
| GND\* | A5 |  |  |  | 3 |  |
| GND\* | A6 |  |  |  | 3 |  |
| GND\* | A7 |  |  |  | 3 |  |
| GND\* | A8 |  |  |  | 3 |  |
| GND\* | A9 |  |  |  | 3 |  |
| GND\* | A10 |  |  |  | 3 |  |
| GND\* | A11 |  |  |  | 3 |  |
| GND+ | A12 |  |  |  | 4 |  |
| GND\* | A13 |  |  |  | 4 |  |
| GND\* | A14 |  |  |  | 4 |  |
| GND\* | A15 |  |  |  | 4 |  |
| GND\* | A16 |  |  |  | 4 |  |
| GND\* | A17 |  |  |  | 4 |  |
| GND\* | A18 |  |  |  | 4 |  |
| GND\* | A19 |  |  |  | 4 |  |
| GND\* | A20 |  |  |  | 4 |  |
| VCCIO4 | A21 | power |  | 3.3V | 4 |  |
| GND | A22 | gnd |  |  |  |  |
| VCCIO1 | AA1 | power |  | 3.3V | 1 |  |
| GND | AA2 | gnd |  |  |  |  |
| GND\* | AA3 |  |  |  | 8 |  |
| GND\* | AA4 |  |  |  | 8 |  |
| GND\* | AA5 |  |  |  | 8 |  |
| GND\* | AA6 |  |  |  | 8 |  |
| GND\* | AA7 |  |  |  | 8 |  |
| GND\* | AA8 |  |  |  | 8 |  |
| GND\* | AA9 |  |  |  | 8 |  |
| GND\* | AA10 |  |  |  | 8 |  |
| GND\* | AA11 |  |  |  | 8 |  |
| GND\* | AA12 |  |  |  | 7 |  |
| GND\* | AA13 |  |  |  | 7 |  |
| GND\* | AA14 |  |  |  | 7 |  |
| GND\* | AA15 |  |  |  | 7 |  |
| GND\* | AA16 |  |  |  | 7 |  |
| GND\* | AA17 |  |  |  | 7 |  |
| GND\* | AA18 |  |  |  | 7 |  |
| GND\* | AA19 |  |  |  | 7 |  |
| GND\* | AA20 |  |  |  | 7 |  |
| GND | AA21 | gnd |  |  |  |  |
| VCCIO6 | AA22 | power |  | 3.3V | 6 |  |
| GND | AB1 | gnd |  |  |  |  |
| VCCIO8 | AB2 | power |  | 3.3V | 8 |  |
| GND\* | AB3 |  |  |  | 8 |  |
| GND\* | AB4 |  |  |  | 8 |  |
| GND\* | AB5 |  |  |  | 8 |  |
| GND\* | AB6 |  |  |  | 8 |  |
| GND\* | AB7 |  |  |  | 8 |  |
| GND\* | AB8 |  |  |  | 8 |  |
| GND\* | AB9 |  |  |  | 8 |  |
| GND\* | AB10 |  |  |  | 8 |  |
| GND\* | AB11 |  |  |  | 8 |  |
| GND\* | AB12 |  |  |  | 7 |  |
| GND\* | AB13 |  |  |  | 7 |  |
| GND\* | AB14 |  |  |  | 7 |  |
| GND\* | AB15 |  |  |  | 7 |  |
| GND\* | AB16 |  |  |  | 7 |  |
| GND\* | AB17 |  |  |  | 7 |  |
| GND\* | AB18 |  |  |  | 7 |  |
| GND\* | AB19 |  |  |  | 7 |  |
| GND\* | AB20 |  |  |  | 7 |  |
| VCCIO7 | AB21 | power |  | 3.3V | 7 |  |
| GND | AB22 | gnd |  |  |  |  |
| VCCIO2 | B1 | power |  | 3.3V | 2 |  |
| GND | B2 | gnd |  |  |  |  |
| GND\* | B3 |  |  |  | 3 |  |
| GND\* | B4 |  |  |  | 3 |  |
| GND\* | B5 |  |  |  | 3 |  |
| GND\* | B6 |  |  |  | 3 |  |
| GND\* | B7 |  |  |  | 3 |  |
| GND\* | B8 |  |  |  | 3 |  |
| GND\* | B9 |  |  |  | 3 |  |
| GND\* | B10 |  |  |  | 3 |  |
| GND\* | B11 |  |  |  | 3 |  |
| GND+ | B12 |  |  |  | 4 |  |
| GND\* | B13 |  |  |  | 4 |  |
| GND\* | B14 |  |  |  | 4 |  |
| GND\* | B15 |  |  |  | 4 |  |
| GND\* | B16 |  |  |  | 4 |  |
| GND\* | B17 |  |  |  | 4 |  |
| GND\* | B18 |  |  |  | 4 |  |
| GND\* | B19 |  |  |  | 4 |  |
| GND\* | B20 |  |  |  | 4 |  |
| GND | B21 | gnd |  |  |  |  |
| VCCIO5 | B22 | power |  | 3.3V | 5 |  |
| GND\* | C1 |  |  |  | 2 |  |
| GND\* | C2 |  |  |  | 2 |  |
| ~nCSO~ / RESERVED\_INPUT \_WITH\_WEAK\_PULLUP | C3 | input | 3.3-V LVTTL |  | 2 | N |
| ~ASDO~ / RESERVED\_INPUT \_WITH\_WEAK\_PULLUP | C4 | input | 3.3-V LVTTL |  | 2 | N |
| GND | C5 | gnd |  |  |  |  |
| VCCIO3 | C6 | power |  | 3.3V | 3 |  |
| GND\* | C7 |  |  |  | 3 |  |
| GND | C8 | gnd |  |  |  |  |
| GND\* | C9 |  |  |  | 3 |  |
| GND\* | C10 |  |  |  | 3 |  |
| VCCIO3 | C11 | power |  | 3.3V | 3 |  |
| VCCIO4 | C12 | power |  | 3.3V | 4 |  |
| GND\* | C13 |  |  |  | 4 |  |
| GND\* | C14 |  |  |  | 4 |  |
| GND | C15 | gnd |  |  |  |  |
| GND\* | C16 |  |  |  | 4 |  |
| GND\* | C17 |  |  |  | 4 |  |
| GND\* | C18 |  |  |  | 4 |  |
| GND\* | C19 |  |  |  | 5 |  |
| GND\* | C20 |  |  |  | 5 |  |
| GND\* | C21 |  |  |  | 5 |  |
| GND\* | C22 |  |  |  | 5 |  |
| GND\* | D1 |  |  |  | 2 |  |
| GND\* | D2 |  |  |  | 2 |  |
| GND\* | D3 |  |  |  | 2 |  |
| GND\* | D4 |  |  |  | 2 |  |
| GND\* | D5 |  |  |  | 2 |  |
| GND\* | D6 |  |  |  | 2 |  |
| GND\* | D7 |  |  |  | 3 |  |
| GND\* | D8 |  |  |  | 3 |  |
| GND\* | D9 |  |  |  | 3 |  |
| GND | D10 | gnd |  |  |  |  |
| GND\* | D11 |  |  |  | 3 |  |
| GND+ | D12 |  |  |  | 3 |  |
| GND | D13 | gnd |  |  |  |  |
| GND\* | D14 |  |  |  | 4 |  |
| GND\* | D15 |  |  |  | 4 |  |
| GND\* | D16 |  |  |  | 4 |  |
| VCCIO4 | D17 | power |  | 3.3V | 4 |  |
| GND | D18 | gnd |  |  |  |  |
| GND\* | D19 |  |  |  | 5 |  |
| GND\* | D20 |  |  |  | 5 |  |
| GND\* | D21 |  |  |  | 5 |  |
| GND\* | D22 |  |  |  | 5 |  |
| GND\* | E1 |  |  |  | 2 |  |
| GND\* | E2 |  |  |  | 2 |  |
| GND\* | E3 |  |  |  | 2 |  |
| GND\* | E4 |  |  |  | 2 |  |
| VCCD\_PLL3 | E5 | power |  | 1.2V |  |  |
| VCCA\_PLL3 | E6 | power |  | 1.2V |  |  |
| GND\* | E7 |  |  |  | 3 |  |
| GND\* | E8 |  |  |  | 3 |  |
| GND\* | E9 |  |  |  | 3 |  |
| VCCIO3 | E10 | power |  | 3.3V | 3 |  |
| GND\* | E11 |  |  |  | 3 |  |
| GND+ | E12 |  |  |  | 3 |  |
| VCCIO4 | E13 | power |  | 3.3V | 4 |  |
| GND\* | E14 |  |  |  | 4 |  |
| GND\* | E15 |  |  |  | 4 |  |
| GNDA\_PLL2 | E16 | gnd |  |  |  |  |
| GND\_PLL2 | E17 | gnd |  |  |  |  |
| GND\* | E18 |  |  |  | 5 |  |
| GND\* | E19 |  |  |  | 5 |  |
| GND\* | E20 |  |  |  | 5 |  |
| GND\* | E21 |  |  |  | 5 |  |
| GND\* | E22 |  |  |  | 5 |  |
| GND\* | F1 |  |  |  | 2 |  |
| GND\* | F2 |  |  |  | 2 |  |
| GND\* | F3 |  |  |  | 2 |  |
| GND\* | F4 |  |  |  | 2 |  |
| GND\_PLL3 | F5 | gnd |  |  |  |  |
| GND\_PLL3 | F6 | gnd |  |  |  |  |
| GNDA\_PLL3 | F7 | gnd |  |  |  |  |
| GND\* | F8 |  |  |  | 3 |  |
| GND\* | F9 |  |  |  | 3 |  |
| GND\* | F10 |  |  |  | 3 |  |
| GND\* | F11 |  |  |  | 3 |  |
| GND\* | F12 |  |  |  | 4 |  |
| GND\* | F13 |  |  |  | 4 |  |
| GND\* | F14 |  |  |  | 4 |  |
| GND\* | F15 |  |  |  | 4 |  |
| VCCA\_PLL2 | F16 | power |  | 1.2V |  |  |
| VCCD\_PLL2 | F17 | power |  | 1.2V |  |  |
| GND\_PLL2 | F18 | gnd |  |  |  |  |
| GND | F19 | gnd |  |  |  |  |
| GND\* | F20 |  |  |  | 5 |  |
| GND\* | F21 |  |  |  | 5 |  |
| GND\* | F22 |  |  |  | 5 |  |
| GND\* | G1 |  |  |  | 2 |  |
| GND\* | G2 |  |  |  | 2 |  |
| GND\* | G3 |  |  |  | 2 |  |
| GND | G4 | gnd |  |  |  |  |
| GND\* | G5 |  |  |  | 2 |  |
| GND\* | G6 |  |  |  | 2 |  |
| VCCINT | G7 | power |  | 1.2V |  |  |
| VCCINT | G8 | power |  | 1.2V |  |  |
| VCCIO3 | G9 | power |  | 3.3V | 3 |  |
| GND | G10 | gnd |  |  |  |  |
| GND | G11 | gnd |  |  |  |  |
| VCCINT | G12 | power |  | 1.2V |  |  |
| GND | G13 | gnd |  |  |  |  |
| VCCIO4 | G14 | power |  | 3.3V | 4 |  |
| GND | G15 | gnd |  |  |  |  |
| VCCINT | G16 | power |  | 1.2V |  |  |
| GND\* | G17 |  |  |  | 5 |  |
| GND\* | G18 |  |  |  | 5 |  |
| VCCIO5 | G19 | power |  | 3.3V | 5 |  |
| GND\* | G20 |  |  |  | 5 |  |
| GND\* | G21 |  |  |  | 5 |  |
| GND\* | G22 |  |  |  | 5 |  |
| GND\* | H1 |  |  |  | 2 |  |
| GND\* | H2 |  |  |  | 2 |  |
| GND\* | H3 |  |  |  | 2 |  |
| GND\* | H4 |  |  |  | 2 |  |
| GND\* | H5 |  |  |  | 2 |  |
| GND\* | H6 |  |  |  | 2 |  |
| VCCINT | H7 | power |  | 1.2V |  |  |
| VCCINT | H8 | power |  | 1.2V |  |  |
| GND | H9 | gnd |  |  |  |  |
| GND | H10 | gnd |  |  |  |  |
| GND | H11 | gnd |  |  |  |  |
| VCCINT | H12 | power |  | 1.2V |  |  |
| VCCINT | H13 | power |  | 1.2V |  |  |
| GND | H14 | gnd |  |  |  |  |
| GND | H15 | gnd |  |  |  |  |
| VCCINT | H16 | power |  | 1.2V |  |  |
| GND\* | H17 |  |  |  | 5 |  |
| GND\* | H18 |  |  |  | 5 |  |
| GND\* | H19 |  |  |  | 5 |  |
| GND | H20 | gnd |  |  |  |  |
| GND\* | H21 |  |  |  | 5 |  |
| GND\* | H22 |  |  |  | 5 |  |
| GND\* | J1 |  |  |  | 2 |  |
| GND\* | J2 |  |  |  | 2 |  |
| GND\* | J3 |  |  |  | 2 |  |
| GND\* | J4 |  |  |  | 2 |  |
| GND\* | J5 |  |  |  | 2 |  |
| GND\* | J6 |  |  |  | 2 |  |
| VCCIO2 | J7 | power |  | 3.3V | 2 |  |
| GND | J8 | gnd |  |  |  |  |
| GND | J9 | gnd |  |  |  |  |
| VCCINT | J10 | power |  | 1.2V |  |  |
| VCCINT | J11 | power |  | 1.2V |  |  |
| VCCINT | J12 | power |  | 1.2V |  |  |
| VCCINT | J13 | power |  | 1.2V |  |  |
| GND | J14 | gnd |  |  |  |  |
| GND | J15 | gnd |  |  |  |  |
| VCCIO5 | J16 | power |  | 3.3V | 5 |  |
| GND\* | J17 |  |  |  | 5 |  |
| GND\* | J18 |  |  |  | 5 |  |
| GND\* | J19 |  |  |  | 5 |  |
| GND\* | J20 |  |  |  | 5 |  |
| GND\* | J21 |  |  |  | 5 |  |
| GND\* | J22 |  |  |  | 5 |  |
| nCE | K1 |  |  |  | 2 |  |
| TCK | K2 | input |  |  | 2 |  |
| GND | K3 | gnd |  |  |  |  |
| DATA0 | K4 | input |  |  | 2 |  |
| TDI | K5 | input |  |  | 2 |  |
| TMS | K6 | input |  |  | 2 |  |
| GND | K7 | gnd |  |  |  |  |
| VCCINT | K8 | power |  | 1.2V |  |  |
| VCCINT | K9 | power |  | 1.2V |  |  |
| GND | K10 | gnd |  |  |  |  |
| GND | K11 | gnd |  |  |  |  |
| GND | K12 | gnd |  |  |  |  |
| GND | K13 | gnd |  |  |  |  |
| VCCINT | K14 | power |  | 1.2V |  |  |
| GND | K15 | gnd |  |  |  |  |
| GND | K16 | gnd |  |  |  |  |
| GND\* | K17 |  |  |  | 5 |  |
| GND\* | K18 |  |  |  | 5 |  |
| GND | K19 | gnd |  |  |  |  |
| GND\* | K20 |  |  |  | 5 |  |
| GND\* | K21 |  |  |  | 5 |  |
| GND\* | K22 |  |  |  | 5 |  |
| GND+ | L1 |  |  |  | 2 |  |
| GND+ | L2 |  |  |  | 2 |  |
| VCCIO2 | L3 | power |  | 3.3V | 2 |  |
| nCONFIG | L4 |  |  |  | 2 |  |
| TDO | L5 | output |  |  | 2 |  |
| DCLK | L6 |  |  |  | 2 |  |
| VCCINT | L7 | power |  | 1.2V |  |  |
| VCCINT | L8 | power |  | 1.2V |  |  |
| VCCINT | L9 | power |  | 1.2V |  |  |
| GND | L10 | gnd |  |  |  |  |
| GND | L11 | gnd |  |  |  |  |
| GND | L12 | gnd |  |  |  |  |
| GND | L13 | gnd |  |  |  |  |
| VCCINT | L14 | power |  | 1.2V |  |  |
| GND | L15 | gnd |  |  |  |  |
| VCCINT | L16 | power |  | 1.2V |  |  |
| GND\* | L17 |  |  |  | 5 |  |
| GND\* | L18 |  |  |  | 5 |  |
| GND\* | L19 |  |  |  | 5 |  |
| VCCIO5 | L20 | power |  | 3.3V | 5 |  |
| GND+ | L21 |  |  |  | 5 |  |
| GND+ | L22 |  |  |  | 5 |  |
| clk | M1 | input | 3.3-V LVTTL |  | 1 | Y |
| reset | M2 | input | 3.3-V LVTTL |  | 1 | Y |
| VCCIO1 | M3 | power |  | 3.3V | 1 |  |
| GND | M4 | gnd |  |  |  |  |
| res[17] | M5 | output | 3.3-V LVTTL |  | 1 | Y |
| res[19] | M6 | output | 3.3-V LVTTL |  | 1 | Y |
| GND | M7 | gnd |  |  |  |  |
| GND | M8 | gnd |  |  |  |  |
| VCCINT | M9 | power |  | 1.2V |  |  |
| GND | M10 | gnd |  |  |  |  |
| GND | M11 | gnd |  |  |  |  |
| GND | M12 | gnd |  |  |  |  |
| GND | M13 | gnd |  |  |  |  |
| VCCINT | M14 | power |  | 1.2V |  |  |
| VCCINT | M15 | power |  | 1.2V |  |  |
| GND | M16 | gnd |  |  |  |  |
| MSEL0 | M17 |  |  |  | 6 |  |
| GND\* | M18 |  |  |  | 6 |  |
| GND\* | M19 |  |  |  | 6 |  |
| VCCIO6 | M20 | power |  | 3.3V | 6 |  |
| GND+ | M21 |  |  |  | 6 |  |
| GND+ | M22 |  |  |  | 6 |  |
| res[21] | N1 | output | 3.3-V LVTTL |  | 1 | Y |
| res[24] | N2 | output | 3.3-V LVTTL |  | 1 | Y |
| res[12] | N3 | output | 3.3-V LVTTL |  | 1 | Y |
| res[11] | N4 | output | 3.3-V LVTTL |  | 1 | Y |
| res[18] | N5 | output | 3.3-V LVTTL |  | 1 | Y |
| res[27] | N6 | output | 3.3-V LVTTL |  | 1 | Y |
| GND | N7 | gnd |  |  |  |  |
| GND | N8 | gnd |  |  |  |  |
| VCCINT | N9 | power |  | 1.2V |  |  |
| GND | N10 | gnd |  |  |  |  |
| GND | N11 | gnd |  |  |  |  |
| GND | N12 | gnd |  |  |  |  |
| GND | N13 | gnd |  |  |  |  |
| VCCINT | N14 | power |  | 1.2V |  |  |
| VCCINT | N15 | power |  | 1.2V |  |  |
| GND | N16 | gnd |  |  |  |  |
| MSEL1 | N17 |  |  |  | 6 |  |
| CONF\_DONE | N18 |  |  |  | 6 |  |
| GND | N19 | gnd |  |  |  |  |
| nSTATUS | N20 |  |  |  | 6 |  |
| GND\* | N21 |  |  |  | 6 |  |
| GND\* | N22 |  |  |  | 6 |  |
| res[20] | P1 | output | 3.3-V LVTTL |  | 1 | Y |
| res[22] | P2 | output | 3.3-V LVTTL |  | 1 | Y |
| res[14] | P3 | output | 3.3-V LVTTL |  | 1 | Y |
| res[7] | P4 | output | 3.3-V LVTTL |  | 1 | Y |
| res[15] | P5 | output | 3.3-V LVTTL |  | 1 | Y |
| res[3] | P6 | output | 3.3-V LVTTL |  | 1 | Y |
| VCCIO1 | P7 | power |  | 3.3V | 1 |  |
| GND | P8 | gnd |  |  |  |  |
| GND | P9 | gnd |  |  |  |  |
| VCCINT | P10 | power |  | 1.2V |  |  |
| VCCINT | P11 | power |  | 1.2V |  |  |
| VCCINT | P12 | power |  | 1.2V |  |  |
| VCCINT | P13 | power |  | 1.2V |  |  |
| VCCINT | P14 | power |  | 1.2V |  |  |
| GND | P15 | gnd |  |  |  |  |
| VCCIO6 | P16 | power |  | 3.3V | 6 |  |
| GND\* | P17 |  |  |  | 6 |  |
| GND\* | P18 |  |  |  | 6 |  |
| GND\* | P19 |  |  |  | 6 |  |
| GND\* | P20 |  |  |  | 6 |  |
| GND\* | P21 |  |  |  | 6 |  |
| GND\* | P22 |  |  |  | 6 |  |
| res[31] | R1 | output | 3.3-V LVTTL |  | 1 | Y |
| res[29] | R2 | output | 3.3-V LVTTL |  | 1 | Y |
| GND | R3 | gnd |  |  |  |  |
| res[23] | R4 | output | 3.3-V LVTTL |  | 1 | Y |
| res[9] | R5 | output | 3.3-V LVTTL |  | 1 | Y |
| res[8] | R6 | output | 3.3-V LVTTL |  | 1 | Y |
| VCCINT | R7 | power |  | 1.2V |  |  |
| VCCINT | R8 | power |  | 1.2V |  |  |
| GND | R9 | gnd |  |  |  |  |
| VCCINT | R10 | power |  | 1.2V |  |  |
| GND | R11 | gnd |  |  |  |  |
| VCCINT | R12 | power |  | 1.2V |  |  |
| GND | R13 | gnd |  |  |  |  |
| GND | R14 | gnd |  |  |  |  |
| VCCINT | R15 | power |  | 1.2V |  |  |
| VCCINT | R16 | power |  | 1.2V |  |  |
| GND\* | R17 |  |  |  | 6 |  |
| GND\* | R18 |  |  |  | 6 |  |
| GND\* | R19 |  |  |  | 6 |  |
| GND\* | R20 |  |  |  | 6 |  |
| GND\* | R21 |  |  |  | 6 |  |
| GND\* | R22 |  |  |  | 6 |  |
| res[1] | T1 | output | 3.3-V LVTTL |  | 1 | Y |
| res[13] | T2 | output | 3.3-V LVTTL |  | 1 | Y |
| res[2] | T3 | output | 3.3-V LVTTL |  | 1 | Y |
| VCCIO1 | T4 | power |  | 3.3V | 1 |  |
| res[5] | T5 | output | 3.3-V LVTTL |  | 1 | Y |
| res[25] | T6 | output | 3.3-V LVTTL |  | 1 | Y |
| VCCINT | T7 | power |  | 1.2V |  |  |
| VCCINT | T8 | power |  | 1.2V |  |  |
| VCCIO8 | T9 | power |  | 3.3V | 8 |  |
| GND | T10 | gnd |  |  |  |  |
| GND | T11 | gnd |  |  |  |  |
| VCCINT | T12 | power |  | 1.2V |  |  |
| GND | T13 | gnd |  |  |  |  |
| VCCIO7 | T14 | power |  | 3.3V | 7 |  |
| VCCINT | T15 | power |  | 1.2V |  |  |
| GND | T16 | gnd |  |  |  |  |
| GND\_PLL4 | T17 | gnd |  |  |  |  |
| GND\* | T18 |  |  |  | 6 |  |
| VCCIO6 | T19 | power |  | 3.3V | 6 |  |
| GND | T20 | gnd |  |  |  |  |
| GND\* | T21 |  |  |  | 6 |  |
| GND\* | T22 |  |  |  | 6 |  |
| res[10] | U1 | output | 3.3-V LVTTL |  | 1 | Y |
| res[16] | U2 | output | 3.3-V LVTTL |  | 1 | Y |
| res[26] | U3 | output | 3.3-V LVTTL |  | 1 | Y |
| GND\* | U4 |  |  |  | 1 |  |
| GND\_PLL1 | U5 | gnd |  |  |  |  |
| VCCD\_PLL1 | U6 | power |  | 1.2V |  |  |
| VCCA\_PLL1 | U7 | power |  | 1.2V |  |  |
| GND\* | U8 |  |  |  | 8 |  |
| GND\* | U9 |  |  |  | 8 |  |
| GND\* | U10 |  |  |  | 8 |  |
| GND+ | U11 |  |  |  | 8 |  |
| GND+ | U12 |  |  |  | 8 |  |
| GND\* | U13 |  |  |  | 7 |  |
| GND\* | U14 |  |  |  | 7 |  |
| GND\* | U15 |  |  |  | 7 |  |
| VCCA\_PLL4 | U16 | power |  | 1.2V |  |  |
| VCCD\_PLL4 | U17 | power |  | 1.2V |  |  |
| GND\* | U18 |  |  |  | 6 |  |
| GND\* | U19 |  |  |  | 6 |  |
| GND\* | U20 |  |  |  | 6 |  |
| GND\* | U21 |  |  |  | 6 |  |
| GND\* | U22 |  |  |  | 6 |  |
| res[4] | V1 | output | 3.3-V LVTTL |  | 1 | Y |
| res[30] | V2 | output | 3.3-V LVTTL |  | 1 | Y |
| GND | V3 | gnd |  |  |  |  |
| GND\* | V4 |  |  |  | 1 |  |
| GND\_PLL1 | V5 | gnd |  |  |  |  |
| GND | V6 | gnd |  |  |  |  |
| GNDA\_PLL1 | V7 | gnd |  |  |  |  |
| GND\* | V8 |  |  |  | 8 |  |
| GND\* | V9 |  |  |  | 8 |  |
| VCCIO8 | V10 | power |  | 3.3V | 8 |  |
| GND\* | V11 |  |  |  | 8 |  |
| GND+ | V12 |  |  |  | 7 |  |
| VCCIO7 | V13 | power |  | 3.3V | 7 |  |
| GND\* | V14 |  |  |  | 7 |  |
| GND\* | V15 |  |  |  | 7 |  |
| GNDA\_PLL4 | V16 | gnd |  |  |  |  |
| GND | V17 | gnd |  |  |  |  |
| GND\_PLL4 | V18 | gnd |  |  |  |  |
| GND\* | V19 |  |  |  | 6 |  |
| GND\* | V20 |  |  |  | 6 |  |
| GND\* | V21 |  |  |  | 6 |  |
| GND\* | V22 |  |  |  | 6 |  |
| res[6] | W1 | output | 3.3-V LVTTL |  | 1 | Y |
| res[28] | W2 | output | 3.3-V LVTTL |  | 1 | Y |
| GND\* | W3 |  |  |  | 1 |  |
| GND\* | W4 |  |  |  | 1 |  |
| GND\* | W5 |  |  |  | 1 |  |
| VCCIO8 | W6 | power |  | 3.3V | 8 |  |
| GND\* | W7 |  |  |  | 8 |  |
| GND\* | W8 |  |  |  | 8 |  |
| GND\* | W9 |  |  |  | 8 |  |
| GND | W10 | gnd |  |  |  |  |
| GND\* | W11 |  |  |  | 8 |  |
| GND+ | W12 |  |  |  | 7 |  |
| GND | W13 | gnd |  |  |  |  |
| GND\* | W14 |  |  |  | 7 |  |
| GND\* | W15 |  |  |  | 7 |  |
| GND\* | W16 |  |  |  | 7 |  |
| VCCIO7 | W17 | power |  | 3.3V | 7 |  |
| GND\* | W18 |  |  |  | 6 |  |
| GND | W19 | gnd |  |  |  |  |
| ~LVDS142p/nCEO~ | W20 | output | 3.3-V LVTTL |  | 6 | N |
| GND\* | W21 |  |  |  | 6 |  |
| GND\* | W22 |  |  |  | 6 |  |
| res[0] | Y1 | output | 3.3-V LVTTL |  | 1 | Y |
| GND\* | Y2 |  |  |  | 1 |  |
| GND\* | Y3 |  |  |  | 1 |  |
| GND\* | Y4 |  |  |  | 1 |  |
| GND\* | Y5 |  |  |  | 8 |  |
| GND\* | Y6 |  |  |  | 8 |  |
| GND\* | Y7 |  |  |  | 8 |  |
| GND | Y8 | gnd |  |  |  |  |
| GND\* | Y9 |  |  |  | 8 |  |
| GND\* | Y10 |  |  |  | 8 |  |
| VCCIO8 | Y11 | power |  | 3.3V | 8 |  |
| VCCIO7 | Y12 | power |  | 3.3V | 7 |  |
| GND\* | Y13 |  |  |  | 7 |  |
| GND\* | Y14 |  |  |  | 7 |  |
| GND | Y15 | gnd |  |  |  |  |
| GND\* | Y16 |  |  |  | 7 |  |
| GND\* | Y17 |  |  |  | 7 |  |
| GND\* | Y18 |  |  |  | 6 |  |
| GND\* | Y19 |  |  |  | 6 |  |
| GND\* | Y20 |  |  |  | 6 |  |
| GND\* | Y21 |  |  |  | 6 |  |
| GND\* | Y22 |  |  |  | 6 |  |

### Usage details:

Total logic elements: 38,681 / 50,528 (77 %)

Total combinational functions: 31,269 / 50,528 (62 %)

Dedicated logic registers: 25,245 / 50,528 (50 %)

Total pins: 34 / 294 (12 %)

Total memory bits: 28,704 / 594,432 (5 %)