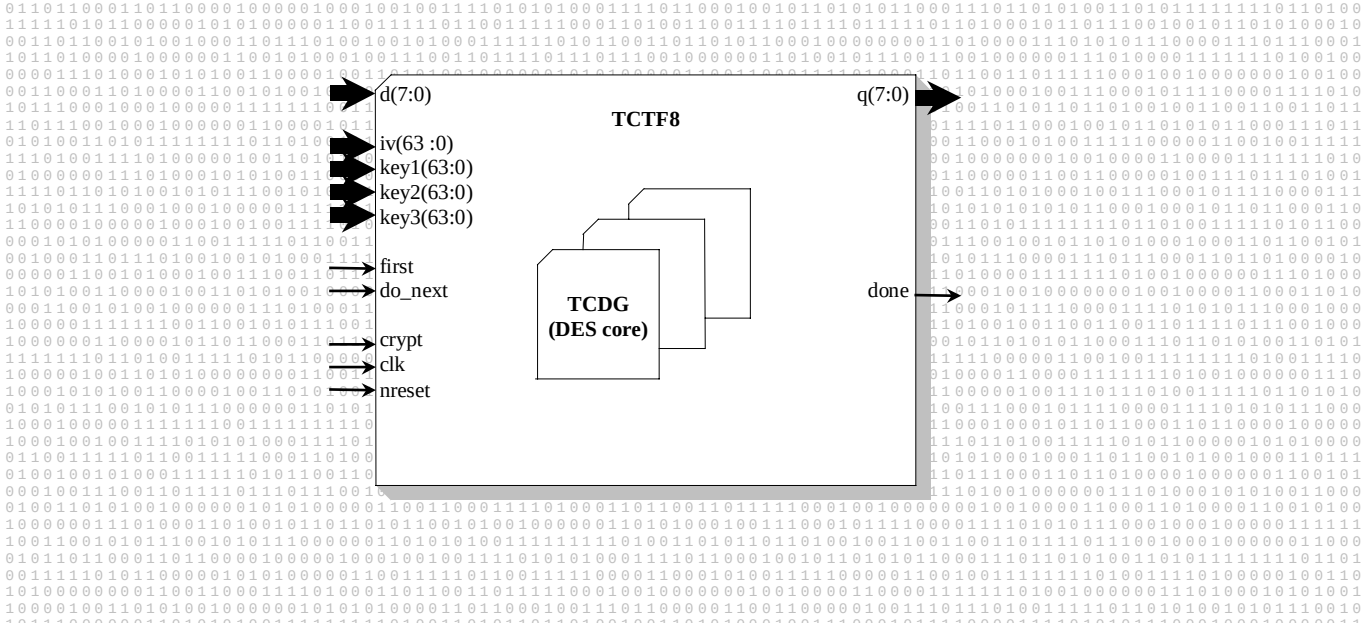
Final

TCTF8

VHDL & Verilog Synthesizable model of the
Triple DES in Cipher Feedback mode, 8-bit data

DISTINCTIVE CHARACTERISTICS

- **High Performance**
  - 48 clock cycles for a complete Triple DES encryption or decryption
  - Simple interface with a start/done handshake
  - No external logic necessary
- **Compatibility**
  - Based on the FIPS PUB 44-2 specification
  - ANSI X3.92, ANSI X9.52
- **Description language & Synthesis caracteristics**
  - Available in VHDL and Verilog
  - Described for both synthesis and simulation
  - Fully Synchronous design
  - Low gate count
  - High clock speed
  - Test bench provided
- **Target Technology**
  - FPGA
  - ASIC
  - Gate Array
  - ...
- **Typical application**
  - Data files protection on any media (hard disk, CD-ROM, EEPROM,...)
  - Access authentification
  - Smart card applications
  - Internet & Intranet communication protection
  - Space telecommunication
  - Banking applications
  - Private informations protections
- **Complete product range**
  - See TETRAEDRE's products portofolio for availability of low-power, full-scan products. As well as for C (C++) source files and DES modules.



Version : 1.0
Printing date : 10/09/99

GENERAL DESCRIPTION

In general, cryptography is used to protect data while it is being communicated between two points or while it is stored on a medium vulnerable to physical theft. Communication security provides protection to data by enciphering it at the transmitting point and deciphering it at the receiving point. File security proceeds protection to data by enciphering it when it is recorded on a storage medium and deciphering it when it is read back from the storage medium. In the first case, the key must be available at the transmitter and receiver simultaneously during communication. In the second case, the key must be maintained and accessible for the duration of the storage period.

The Data Encryption Standard (DES) algorithm, adopted by the U.S. government in 1977, is a block cipher that transforms 64-bit data blocks under a 56-bit secret key, by means of permutation and substitution. It is officially described in FIPS PUB 46. The DES algorithm is used for many applications within the government and in the private sector.

Even through the data encryption standard (DES) provides a high level of security (there are more than 70 quadrillion of possible keys), some applications may demand higher levels of security. The triple DES (TDES) algorithm implemented by this product provides the required security.

Keys
Three 64-bit keys must be provided to the TCTF8 module (192 bits in total). Each key is used by each of the three DES operation performed in the algorithm.
A key consists of 64 binary digits ("0"s or "1"s) of which 56 bits are randomly generated and used directly by the algorithm. The other 8 bits, which are not used by the algorithm may be used for error detection if they are interpreted as parity bits. The DES core module doesn't use these bits at all. When these parity bit are used, they must be set to make the parity of each 8-bit byte of the key odd.
The user must have the key that was used to encipher the data in order to decrypt it. Use of a different key causes the cipher that is produced for any given set of inputs to be different. The encryption algorithm specified for the DES is commonly known among those using the standard. The cryptographic security of the data depends on the security provided for the key used to encipher and decipher the data.

Input Vector
Since the result of a TDES operation depends on the result of the preceding calculation, the first calculation must be initialized using an "initial vector" (IV). This initial vector is a 64-bit vector composed of binary digits ("0"s and "1"s).
The user must have the initial vector that was used to encipher the data in order to decrypt it. Use of a different initial vector causes the cipher that is produced for any given set of inputs to be different.

Data input
The TDES algorithm can crypt or decrypt any data (either text, numbers) expressed in 8 binary digits words. Therefore, this algorithm can be used to protect any kind of documents, like pictures, private data, bank account number, confidential documents, ...

Data output
Data can be recovered from cipher only by using exactly the same key and initial vector used to encipher it. Unauthorized recipients of the cipher who know the algorithm but do not have the correct keys cannot derive the original data algorithmically. However, anyone who does have the keys can decipher the cipher and obtain the original data.

Qualification

The cryptographic DES algorithm transforms a 64-bit binary value into a unique 64-bit binary value based on a 56-bit variable. If the complete 64-bit input is used and if the 56-bit variable is randomly chose, no technique other than trying all possible keys using known input and output for the DES will guarantee finding the chosen key. As these are over 70,000,000,000,000,000 (seventy quadrillion) possible keys of 56 bits, the feasibility of deriving a particular key in this way is extremely unlikely in typical threat environments. Moreover, if the key is changed frequently, the risk of this event is greatly diminished.

Since the TDES algorithm uses three consecutive DES operations, the security level of the coding in therefore drastically increased.

Export control and restrictions

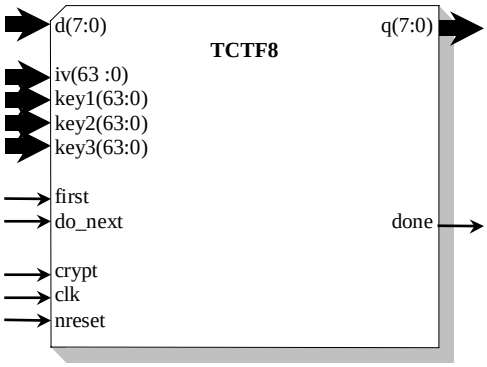
In the United States of America, cryptographic devices and technical data regarding them are subject to U.S. Federal Government export controls (Code of Federal Regulations). Some export of cryptographic modules implementing this standard and technical data regarding them must comply with these regulations and be licensed by the U.S. Department of State or by the Bureau of Export Administration of the U.S. Department of Commerce.

This product is not suitable for life support equipment.

Patents

Cryptographic devices implementing this standard may be covered by U.S. and foreign patents issued to the International Business Machines Corporation. However, IBM has granted nonexclusive, royalty-free licenses under the patents to make, use and sell apparatus which complies with this standard.

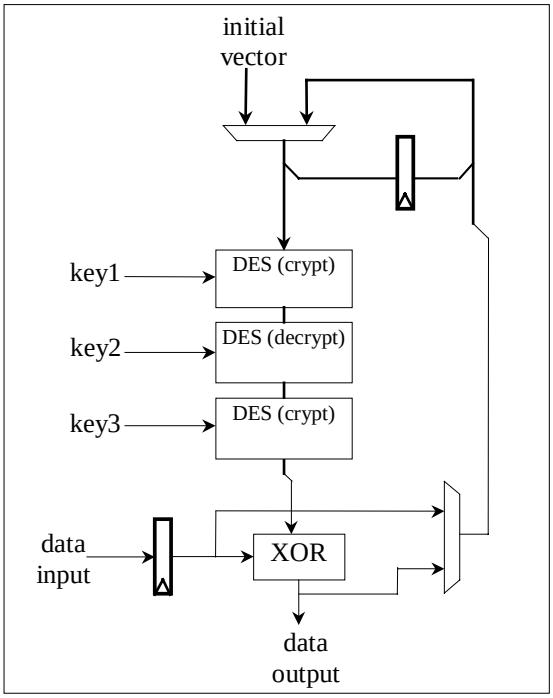
LOGIC SYMBOL



PIN CONFIGURATION

d(7:0)	data input
q(7:0)	result output
iv(63:0)	Initial vector
key1(63:0)	key #1 input
key2(63:0)	key #2 input
key3(63:0)	key #3 input
first	First of sequence notification
do_next	conversion start
crypt	crypt/decrypt mode selection
done	conversion done flag
clk	clock input
nreset	asynchronous reset

Block diagram



OPERATING MODES

The TCTF8 module is a very easy to use component. The data to be encrypted or decrypted is applied to the **d** input, the appropriate operation mode is selected by setting **crypt** with the appropriate value. Then, to start the conversion, the **do_next** signal must be asserted. Once the conversion is finished, the **done** signal is asserted. The result can be read on the **q** output pin.
The word "conversion" is used hereafter to indicate either an encryption or a decryption. The choice of the operation is selected by the crypt pin, as explained below.

To improve the security of coding, the TDES algorithm uses the result of the previous TDES operation to calculate the result of the current operation. To indicate the first operation of the sequence, the **first** signal must be asserted at the beginning of the calculation.

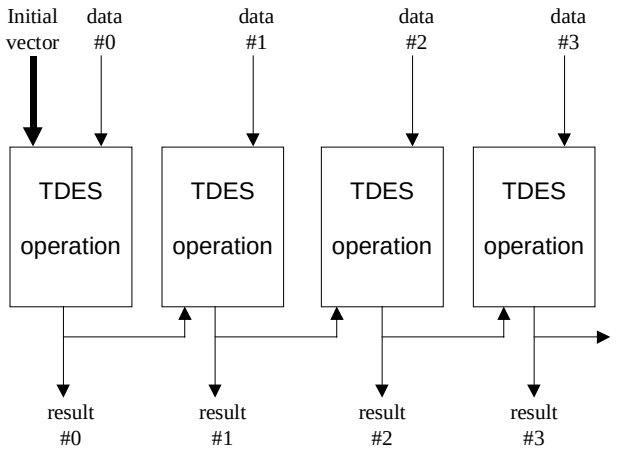


Figure 1. TDES operations series

A register is implemented inside the TCTF8 module to store the result of the previous operation, thus simplifing the use of the module.

Since the design is completely synchronous, no clock signals are generated and no gated clock are used inside this component. So the interface between the DES module and your application is very easy to implement.

Pinout description

key1(63:0) This 64-bit wide bus represents the key for the first DES calculation. The key input must remain stable during the complete conversion. The key must not change until the result has been stored since the output depends directly on the key value.

key2(63:0) This 64-bit wide bus represents the key for the second DES calculation. The key input must remain stable during the complete conversion. The key must not change until the result has been stored since the output depends directly on the key value.

key3(63:0) This 64-bit wide bus represents the key for the third DES calculation. The key input must remain stable during the complete conversion. The key must not change until the result has been stored since the output depends directly on the key value.

IV(63:0) This 64-bit wide bus represents the initial vector of the TDES process. This input must remain stable during the complete process and *not only during the first conversion* but also during the others since it is used during several conversions.

d(7:0) This 8-bit input represents the data which must be encrypted or decrypted. The data is sampled in internal registers at **clk**'s rising edge at the conversion's start.

q(7:0) This 8-bit output represents the result of the calculation. This signal can be sampled by your application at **clk**'s rising edge, when **done** is asserted (high). The result appears on the output line simultaneously with the **done** signal.

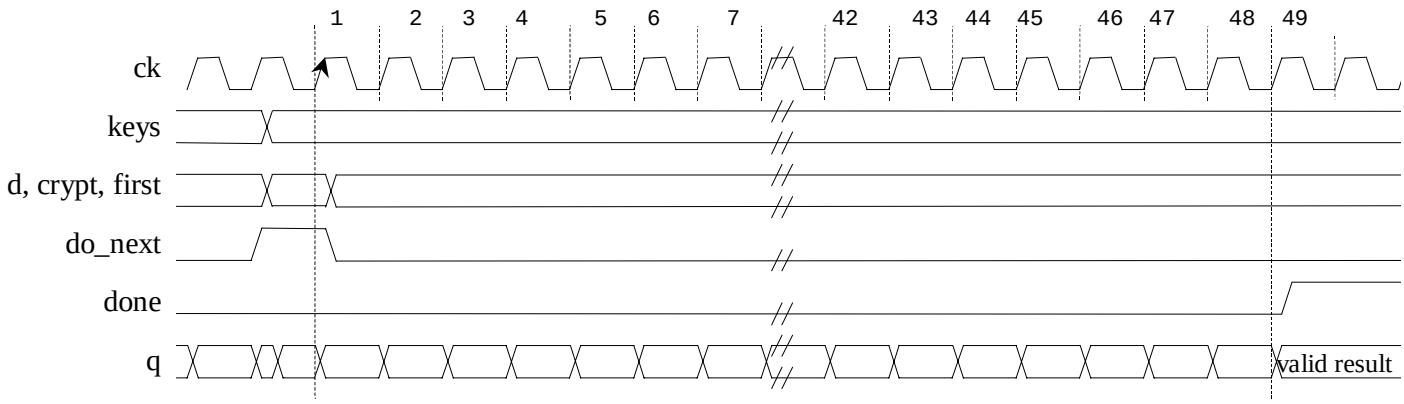


Figure 2. Conversion timing

do_next This signal indicates to the internal state machine to start a conversion. The conversion starts at **clk**'s rising edge after **do_next** has changed from 0 to 1. The **do_next** signal is internally synchronized with the clock.

first This signal indicates if the following conversion in the first conversion of a serie or not.

crypt This signal indicates if the data at the input must be encrypted (crypt=1) or decrypted (crypt=0). This signal is sampled in an internal registers at **clk**'s rising edge when the conversion starts.

done This output indicates (at 1) when the result has been calculated. This signal remains at one as long as no START signal is asserted.

clk Clock signal.

nreset This signal is an asynchronous reset signal (low active).

Conversion timing diagram

The figure 2 shows the timing diagram of a conversion. Since the data (**d**) and crypt inputs are sampled at the start of the conversion, they can change after the start. The key input is not sampled inside the component so modifying the key will change the result, even after the conversion ended.
The **do_next** signal starts the conversion. **done** is asserted 48 clock cycles after start has been negated. The result remains stable as long as no START occurs and as long as the inputs remain stable.

COMPILATION & SIMULATION INFORMATION

The TCTF module instantiates the TCDG component to perform the DES operation. This component is given in a source file (*tcdg.vhd* or *tcdg.v*) which is different of the TCTF's one. It is also provided with its own testbench (*tcdg_bench.vhd* or *tcdg_bench.v*) because it is a very specific function which should be tested separately.

During compilation process, the TCDG component must be compiled before the TCTF component.

The TCTF and TCDG (DES) components are delivered with a HDL testbench (either VHDL or Verilog) for each. These testbench is composed of several hundreds encryption and decryption calculations. The test vectors define the input parameters (d, key, crypt) and also the expected output of the block. The value generated by the DES module (or triple DES) is compared with the expected value. If they don't match, the "erreur" signal is asserted (high) and an error message is displayed on the simulator's standard output[see note 1]. The "erreur" signal is negated at the beginning of the simulation and remains low as long as no error is detected.

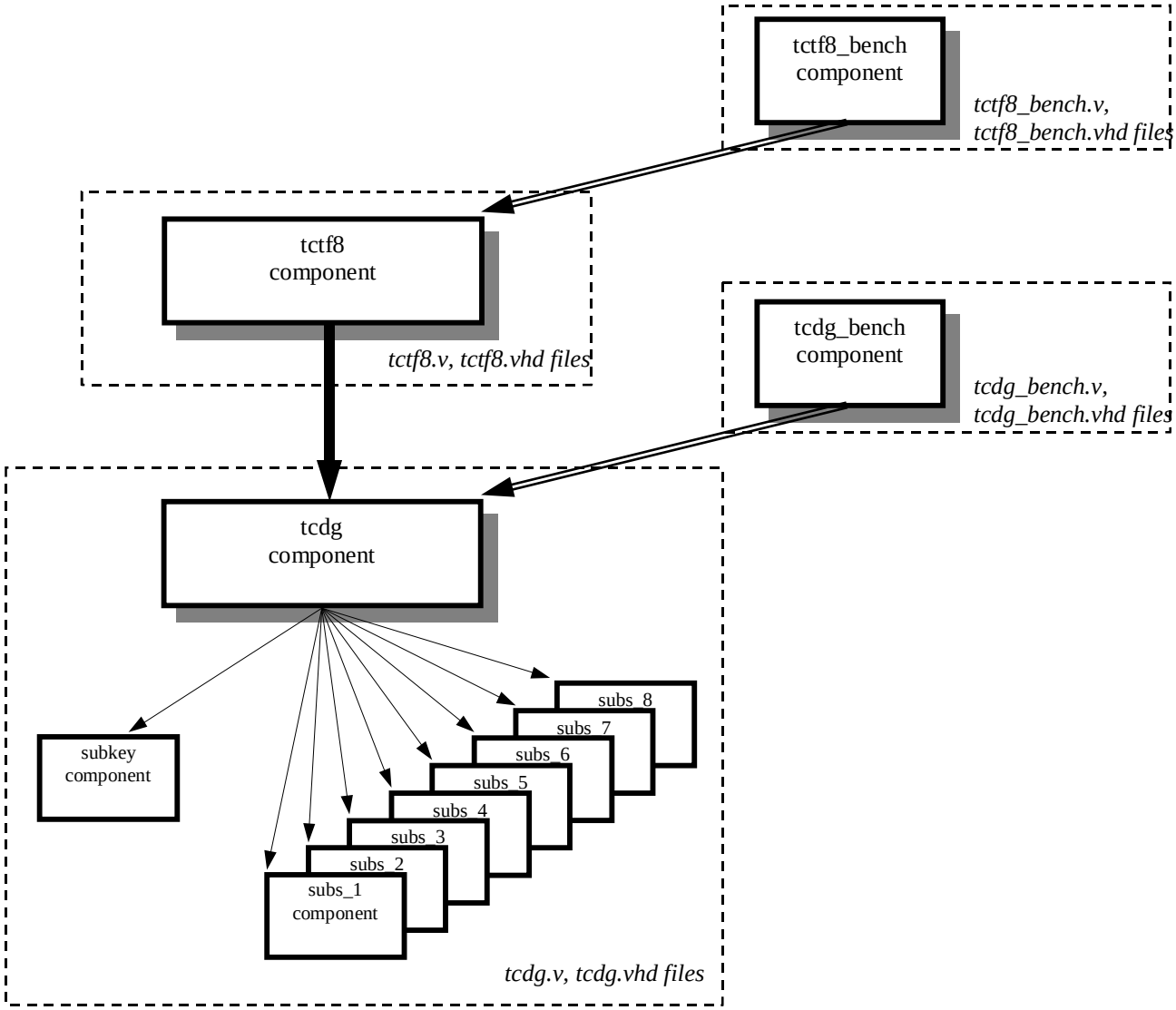


Figure 3. HDL files and design hierachy

1 This feature is only available for the VHDL description. For the Verilog model, the value of the "erreur" signal must be tested at the end of the simulation.

At the end of the simulation, a message is displayed indicating if the complete test failed or if it succeed [see note 2]. This message has a severity of type "error" if the test failed and a severity of type "note" if it succeed. Don't forget to enable the display of these message for a proper simulation.

The TCDG (DES) component and all its sub-blocks are described in a single file: *tcdg.vhd* or *tcdg.v*.
The TCDG's testbench is completely described in the *tcdg_bench.v* or *tcdg_bench.vhd* file.

The TCTF8 component is described in the file: *tctf8.vhd* or *tctf8.v*. Its only sub-block is TCDG and is described in the file *tcdg.vhd* or *tcdg.v* as explained above.

The TCTF8's testbench is completely described in the *tctf8_bench.v* or *tctf8_bench.vhd* file.

Testbench files must be compiled after the block they test since they have the highest level of hierarchy.
The figure 3 shows this hierarchy.

Compilation and Simulation script

The following commands are for Mentor ModelSim Tools™

Library creation:

vlib work

Compilation:

vcom -work work -explicit tcdg.vhd
vcom -work work -explicit tctf8.vhd

Compilation of the test bench:

vcom -work work -explicit tcdg_bench.vhd
vcom -work work -explicit tctf8_bench.vhd

Simulation:

vsim -lib work tctf8_bench

Validation

The TCDG module is validated using test vectors. These vectors are composed of input parameters and an expected result. This expected result is automatically compared with the result of the TCTF8 operation. The DES core (TCDG module) is tested either through the TCTF8's testbench or using a specific testbench (TCDG_TESTBENCH).

The data input and expected results for these testbenches are taken from the following sources:

- The ANSI X3.106 specification
- The ANSI X9.52 specification
- Some vectors have been calculated using shareware using the DES algorithm and compared with the result of the TCDG module.
- Some vectors have been calculated with the C++ description of the DES function.

In addition, the different permutation tables and the S-Box are formally compared with the one in the ANSI specification.

2 This feature is only available for the VHDL description. For the Verilog model, the value of the "erreur" signal must be tested at the end of the simulation.

SYNTHESIS INFORMATION

Architecture

The TCTF8 has a single sub-block; TCDG, the DES core. The DES has nine sub-blocks: "subkey" and "subs_1" to "subs_8". All the sub-blocks are completely combinatorial. The synthesis can be made directly on the top module and flattening will not modify drastically the result in term of area or speed. The DES component and all its sub-blocks are grouped in the same file.
The subs (S-Box permutation) are mainly look-up tables. Due to the specificity of the DES algorithm (pseudo-random permutations), the S-Box cannot be very well optimized.

The internal architecture of the DES is mainly composed of multiplexers, XOR gated and the substitution boxes. The control logic is very simple.

The easiest way to synthesize this component is to define a clock, set the correct input and output constraints regarding your design's constraints and to optimize the block (see "synthesis crypt below)

Basically the DES uses only a 56-bit key, but for most applications, a 64-bit key is provided. The 8 unused bits are often treated as parity control bit.
In this HDL description, the three key inputs are each 64-bit wide therefrom 8 inputs are unused. These unused inputs will be notified by the synthesizer.

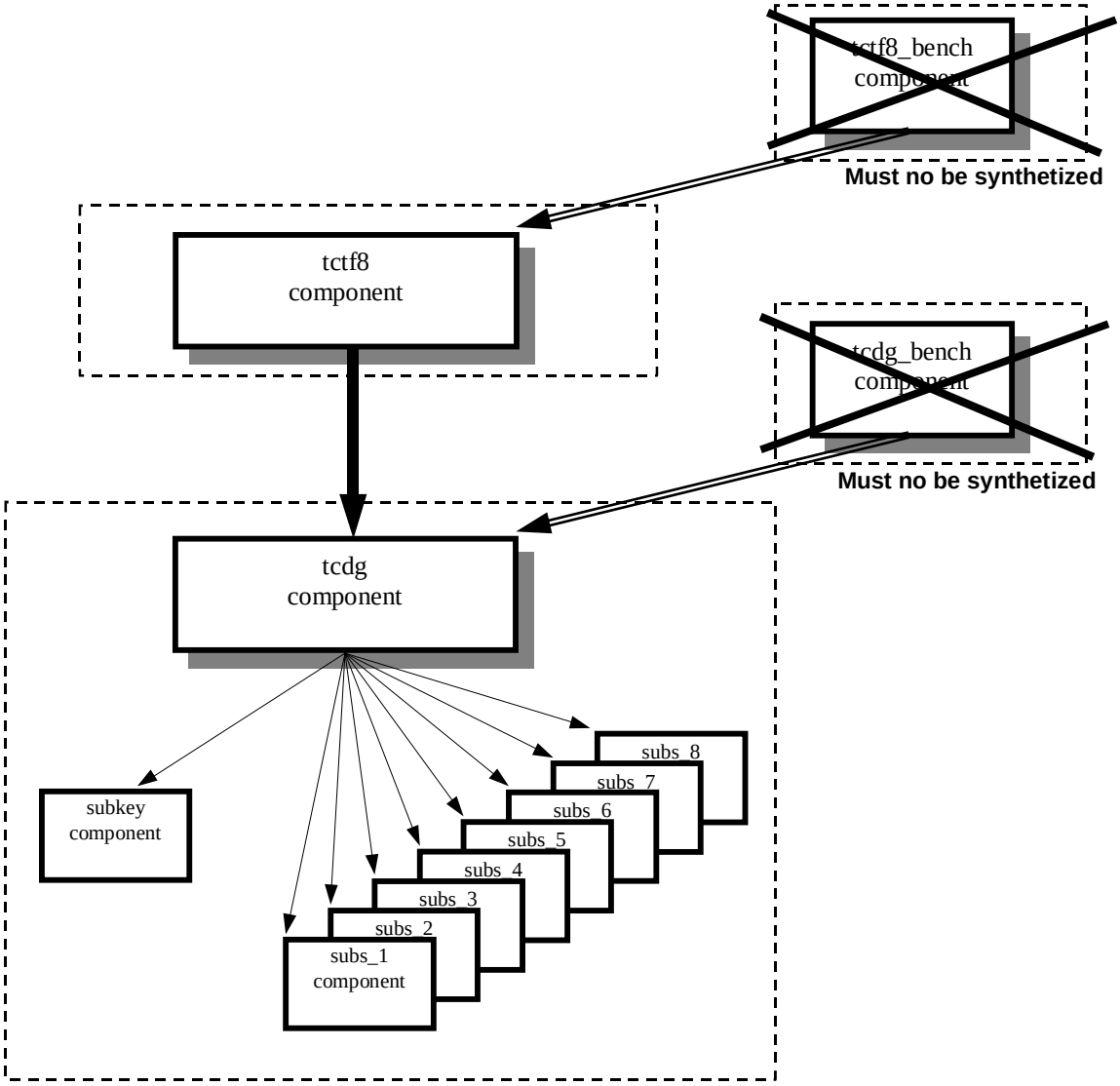


Figure 4. Design hierachy and synthesis

Synthesis

The following commands provide you an example of constraints to synthesize the TCTF8 module with your Synthesizer.

synthesis script (for Synplicity™ Tools):

```
define_clock clk -freq 25.0

define_input_delay  {iv[63:0]}      -1000
define_input_delay {key1[63:0]}     -1000
define_input_delay {key2[63:0]}     -1000
define_input_delay {key3[63:0]}     -1000
define_input_delay {d[7:0]}         10
define_input_delay do_next          7
define_input_delay first            7
define_input_delay crypt            5
define_input_delay nreset           -1000

define_output_delay done            15
define_output_delay {q[7:0]}        5
```

synthesis result on FPGA Actel A54SX32:

Estimated frequency:	28 MHz
IO number:	254
Sequential cells:	288*
Combinatorial cells:	1210

*The number of sequential cells given by the synthesizer is bigger here than the number of necessary flip-flops because of fan-out constraints. (Results given by Synplicity™ Tools).

ORDERING INFORMATION

Worldwide Sales Offices

Tetraedre Sarl
Chenes 19
2072 Saint-Blaise
Switzerland

e-mail: sales@tetraedre.com
web: www.tetraedre.com
phone: +41 79 402 25 39
fax: +41 86 079 402 25 39

Device Number

TCTF8 VHDL	VHDL description of TDES cipher feedback mode 8-bit, TCDG (DES core) functional test benchs.
TCTF8 Verilog	Verilog description of TDES cipher feedback mode 8-bit, TCDG (DES core) functional test benchs.