10.2 ANEXO 2: Sintetización (información FPGA)

```
Design Statistics
# IOs                    : 11
Cell Usage :
# BELS                   : 2817
#      GND               : 9
#      INV               : 74
#      LUT1              : 219
#      LUT2              : 303
#      LUT2_D            : 15
#      LUT2_L            : 1
#      LUT3              : 320
#      LUT3_D            : 11
#      LUT3_L            : 20
#      LUT4              : 796
#      LUT4_D            : 31
#      LUT4_L            : 33
#      MUXCY             : 495
#      MUXCY_L           : 15
#      MUXF5             : 37
#      MUXF6             : 1
#      VCC               : 9
#      XORCY             : 428
# FlipFlops/Latches      : 1347
#      FD                : 109
#      FD_1              : 5
#      FDC               : 50
#      FDCE              : 32
#      FDCPE             : 4
#      FDE               : 12
#      FDP               : 11
#      FDPE              : 5
#      FDR               : 423
#      FDRE              : 615
#      FDRS              : 29
#      FDRSE             : 22
#      FDS               : 19
#      FDSE              : 11
# RAMS                   : 144
#      RAM16X1D          : 110
#      RAMB16_S2_S2      : 32
#      RAMB16_S36_S36    : 2
# Shift Registers        : 57
#      SRL16             : 38
#      SRL16E            : 19
# Clock Buffers          : 1
#      BUFG              : 1
# IO Buffers             : 11
#      IBUF              : 2
#      IBUFG             : 1
#      OBUF              : 8
# DCMs                   : 1
#      DCM               : 1
# Others                 : 1
#      PPC405            : 1
```

Device utilization summary:

Selected Device : 2vp30ff896-7

Number of Slices:	1522	out of 13696	11%
Number of Slice Flip Flops:	1347	out of 27392	4%
Number of 4 input LUTs:	2100	out of 27392	7%
Number used as logic:	1823		
Number used as Shift registers:	57		
Number used as RAMs:	220		
Number of IOs:	11		
Number of bonded IOBs:	11	out of 556	1%
Number of BRAMs:	34	out of 136	25%
Number of GCLKs:	1	out of 16	6%
Number of PPC405s:	1	out of 2	50%
Number of DCMs:	1	out of 8	12%

Clock Information:

Clock Signal	Clock buffer(FF name)	Load
sys_clk_pin	dcm_0/Using_Virtex.DCM_INST:CLK0	1549
fpga_0_net_gnd_pin_OBUF	NONE(plb_bram_if_cntlr_1_bram/plb_bram_if_cntlr_1_bram/ramb16_s2_s2_21)	32

Asynchronous Control Signals Information:

Control Signal	Buffer(FF name)	Load
my_pattern_0/my_pattern_0/OPB_IPIF_I/OPB_BAM_I/INCLUDE_WRFIFO_GEN.I_WRPFIFO_TOP/I_IPIF_INTERFACE_BLOCK/Fifo_Reset(my_pattern_0/my_pattern_0/OPB_IPIF_I/OPB_BAM_I/INCLUDE_WRFIFO_GEN.I_WRPFIFO_TOP/I_IPIF_INTERFACE_BLOCK/Fifo_Reset:Q)	NONE(my_pattern_0/my_pattern_0/OPB_IPIF_I/OPB_BAM_I/INCLUDE_WRFIFO_GEN.I_WRPFIFO_TOP/USE_BLOCK_RAM.I_DP_CONTROLLER/trans_state_FFd3)	16
opb/OPB_Rst(opb/opb/POR_FF_I:Q)	NONE(rs232_uart_1/rs232_uart_1/OPB_UARTLITE_Core_I/OPB_UARTLITE_RX_I/previous_RX)	68
plb2opb/plb2opb/PLBside_reset_OPB_timeout_onRd(plb2opb/plb2opb/I_OPB_timeout_side_Reg_CLR:O)	NONE(plb2opb/plb2opb/I_OPB_timeout_Reg)	1

plb2opb/BGO_dcrDBus<31>(plb2opb/XST_GND:G)
| NONE(plb2opb/plb2opb/I_OPB_timeout_Reg)
| 4 |

rs232_uart_1/rs232_uart_1/OPB_UARTLITE_Core_I/reset_TX_FIFO(rs232_uart_1/rs232_uart_1/OPB_UARTLITE_Core_I/reset_TX_FIFO:Q)
|
NONE(rs232_uart_1/rs232_uart_1/OPB_UARTLITE_Core_I/OPB_UARTLITE_TX_I/SRL_FIFO_I/data_Exists_I) | 1 |

my_pattern_0/my_pattern_0/OPB_IPIF_I/OPB_BAM_I/INCLUDE_RDFIFO_GEN.I_RDFIFO/I_IPIF_INTERFACE_BLOCK/Fifo_rst(my_pattern_0/my_pattern_0/OPB_IPIF_I/OPB_BAM_I/INCLUDE_RDFIFO_GEN.I_RDFIFO/I_IPIF_INTERFACE_BLOCK/Fifo_rst:Q) |
NONE(my_pattern_0/my_pattern_0/OPB_IPIF_I/OPB_BAM_I/INCLUDE_RDFIFO_GEN.I_RDFIFO/USE_BLOCK_RAM.I_DP_CONTROLLER/trans_state_FFd2) | 12 |

plb2opb/plb2opb/PLBside_reset_OPB_retry_onRd(plb2opb/plb2opb/I_B_side_Reg_CLR:O)
| NONE(plb2opb/plb2opb/I_A_side_Reg) | 1
|

plb2opb/plb2opb/PLB_abort_regd_clear(plb2opb/plb2opb/I_PLB_abort_regd_clear:O)
| NONE(plb2opb/plb2opb/I_PLB_abort_Reg) |
1 |

rs232_uart_1/rs232_uart_1/OPB_UARTLITE_Core_I/reset_RX_FIFO(rs232_uart_1/rs232_uart_1/OPB_UARTLITE_Core_I/reset_RX_FIFO:Q)
|
NONE(rs232_uart_1/rs232_uart_1/OPB_UARTLITE_Core_I/OPB_UARTLITE_RX_I/SRL_FIFO_I/data_Exists_I) | 1 |

plb2opb/plb2opb/OPBside_reset_Read_inprog_negedge_regd(plb2opb/plb2opb/I_OPBside_reset_Read_inprog_negedge_regd:O) |
NONE(plb2opb/plb2opb/I_Read_inprog_negedge_Reg)
| 1 |

+---
--------+-------+
Timing Summary:

Speed Grade: -7

Minimum period: 6.819ns (Maximum Frequency: 146.657MHz)
Minimum input arrival time before clock: 1.746ns
Maximum output required time after clock: 3.293ns
Maximum combinational path delay: No path found

Timing Detail:

All values displayed in nanoseconds (ns)
==========
Timing constraint: Default period analysis for Clock 'sys_clk_pin'

Clock period: 6.819ns (frequency: 146.657MHz)
Total number of paths / destination ports: 81123 / 4852

Delay: 6.819ns (Levels of Logic = 9)
Source: plb/plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/PAValid (FF)
Destination: plb/plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/Cnt_on_plb2opb_rearb_6 (FF)

Source Clock: sys_clk_pin rising
Destination Clock: sys_clk_pin rising

Data Path: plb/plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/PAValid to plb/plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/Cnt_on_plb2opb_rearb_6

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
FDR:C->Q	14	0.370	0.688	plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/PAValid (PLB_PAValid)
end scope: 'plb'				
begin scope: 'plb2opb'				
LUT3:I0->O	5	0.275	0.526	plb2opb/PLB_PAValid_int1 (plb2opb/PLB_PAValid_int)
LUT4:I1->O	2	0.275	0.396	plb2opb/PLB_IF_I/access_valid_rst_or000011 (plb2opb/BGO_rearbitrate_int)
LUT3:I2->O	5	0.275	0.564	plb2opb/BGO_rearbitrate1 (BGO_rearbitrate)
end scope: 'plb2opb'				
begin scope: 'plb'				
LUT2:I0->O	7	0.275	0.483	plb/I_PLB_SLAVE_ORS/REARB_OR/Y_0_or00001 (PLB_Srearbitrate)
LUT4:I2->O	1	0.275	0.468	plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/Set_Cnt_on_plb2opb_rearb_en_SW0_SW0 (N650)
LUT4:I0->O	4	0.275	0.451	plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/Set_Cnt_on_plb2opb_rearb_en (plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/Set_Cnt_on_plb2opb_rearb_en)

```
    LUT4:I3->O            4   0.275

   0.413  plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/Cnt_on_plb2opb_rearb_or0000
(plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/Cnt_on_plb2opb_rearb_or0000)

    FDRS:R                0.536
plb/I_PLB_ARBITER_LOGIC/I_ARBCONTROL_SM/Cnt_on_plb2opb_rearb_0

    ----------------------------------------

    Total                 6.819ns (2.831ns logic, 3.988ns route)

                            (41.5% logic, 58.5% route)

==========================================================================

Timing constraint: Default OFFSET IN BEFORE for Clock 'sys_clk_pin'

  Total number of paths / destination ports: 2 / 2
--------------------------------------------------------------------------
Offset:             1.746ns (Levels of Logic = 2)
Source:             sys_rst_pin (PAD)
Destination:        reset_block/reset_block/EXT_LPF/exr_d1 (FF)
Destination Clock: sys_clk_pin rising

Data Path: sys_rst_pin to reset_block/reset_block/EXT_LPF/exr_d1

                          Gate     Net

    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
    IBUF:I->O             1   0.878

   0.331  sys_rst_pin_IBUF (sys_rst_pin_IBUF)

    begin scope: 'reset_block'

    FDR:R                 0.536        reset_block/EXT_LPF/exr_d1

    ----------------------------------------

    Total                 1.746ns (1.414ns logic, 0.331ns route)

                            (81.0% logic, 19.0% route)

==========================================================================

Timing constraint: Default OFFSET OUT AFTER for Clock 'sys_clk_pin'

  Total number of paths / destination ports: 1 / 1
--------------------------------------------------------------------------
Offset:             3.293ns (Levels of Logic = 2)
Source:             rs232_uart_1/rs232_uart_1/OPB_UARTLITE_Core_I/OPB_UARTLITE_TX_I/TX (FF)
Destination:        fpga_0_RS232_Uart_1_TX_pin (PAD)
Source Clock:       sys_clk_pin rising
```

Data Path: rs232_uart_1/rs232_uart_1/OPB_UARTLITE_Core_I/OPB_UARTLITE_TX_I/TX to fpga_0_RS232_Uart_1_TX_pin

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
FDP:C->Q	1	0.370	0.332	rs232_uart_1/OPB_UARTLITE_Core_I/OPB_UARTLITE_TX_I/TX (TX)
end scope: 'rs232_uart_1'				
OBUF:I->O		2.592		fpga_0_RS232_Uart_1_TX_pin_OBUF (fpga_0_RS232_Uart_1_TX_pin)
Total		3.293ns (2.962ns logic, 0.332ns route)		
		(89.9% logic, 10.1% route)		

==========

CPU : 29.31 / 29.37 s | Elapsed : 39.00 / 40.00 s

-->

Total memory usage is 206612 kilobytes

Number of errors : 0 (0 filtered)

Number of warnings : 32 (0 filtered)

Number of infos : 9 (0 filtered)

Design Summary:

Number of errors: 0

Number of warnings: 6

Logic Utilization:

Number of Slice Flip Flops:	1,114 out of 27,392	4%
Number of 4 input LUTs:	1,133 out of 27,392	4%

Logic Distribution:

Number of occupied Slices:	1,148 out of 13,696	8%
Number of Slices containing only related logic:	1,148 out of 1,148	100%
Number of Slices containing unrelated logic:	0 out of 1,148	0%

*See NOTES below for an explanation of the effects of unrelated logic

Total Number of 4 input LUTs:	1,433 out of 27,392	5%
Number used as logic:	1,133	
Number used as a route-thru:	50	
Number used for Dual Port RAMs:	196	
(Two LUTs used per Dual Port RAM)		
Number used as Shift registers:	54	
Number of bonded IOBs:	11 out of 556	1%
IOB Flip Flops:	2	
Number of PPC405s:	1 out of 2	50%
Number of Block RAMs:	33 out of 136	24%
Number of GCLKs:	1 out of 16	6%
Number of DCMs:	1 out of 8	12%

```
Number of GTs:                  0 out of      8   0%
Number of GT10s:                0 out of      0   0%

Total equivalent gate count for design:  2,215,077
Additional JTAG gate count for IOBs:  528
Peak Memory Usage:  239 MB
Total REAL time to MAP completion:  20 secs
Total CPU time to MAP completion:   16 secs

Initializing temperature to 85.000 Celsius. (default - Range: -40.000 to 100.000 Celsius)
Initializing voltage to 1.400 Volts. (default - Range: 1.400 to 1.600 Volts)

Device speed data version:  "PRODUCTION 1.92 2006-10-19".

Device Utilization Summary:

  Number of BUFGMUXs                    1 out of 16     6%
  Number of DCMs                        1 out of 8      12%
  Number of External IOBs               11 out of 556   1%
  Number of LOCed IOBs                  11 out of 11    100%
  Number of PPC405s                     1 out of 2      50%
  Number of RAMB16s                     33 out of 136   24%
  Number of SLICEs                      1148 out of 13696   8%

Overall effort level (-ol):  High
Placer effort level (-pl):  High
Placer cost table entry (-t): 1
Router effort level (-rl):  High

Starting initial Timing Analysis.  REAL time: 10 secs
Finished initial Timing Analysis.  REAL time: 11 secs
**************************
Generating Clock Report
**************************
+---------------------+--------------+------+------+------------+-------------+
|      Clock Net      |   Resource   |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+--------------+------+------+------------+-------------+
|plb_bram_if_cntlr_1_ |              |      |      |            |             |

|     port_BRAM_Clk |     BUFGMUX7S| No   |  855 |  0.268     |  1.250      |

+---------------------+--------------+------+------+------------+-------------+

Timing summary:
---------------
Timing errors: 0  Score: 0
Constraints cover 62334 paths, 0 nets, and 8687 connections

Design statistics:
   Minimum period:   9.799ns (Maximum frequency: 102.051MHz)

powerpc-eabi-size TestApp_Peripheral/executable.elf

   text      data       bss       dec       hex   filename

   5526       356      8232     14114      3722   TestApp_Peripheral/executable.elf
```

'1': Putting device in ISP mode...done.

Maximum TCK operating frequency for this device chain: 15000000.

Validating chain...

Boundary-scan chain validated successfully.

Elapsed time = 1 sec.

Elapsed time = 0 sec.

'3': Loading file 'implementation/download.bit' ...

done.

INFO:iMPACT:501 - '3': Added Device xc2vp30 successfully.

Maximum TCK operating frequency for this device chain: 15000000.
Validating chain...
Boundary-scan chain validated successfully.
'3': Programming device...

done.

'3': Reading status register contents...

CRC error : 0
Decryptor security set : 0
DCM locked : 1
DCI matched : 1
legacy input error : 0
status of GTS_CFG_B : 1
status of GWE : 1
status of GHIGH : 1
value of MODE pin M0 : 1
value of MODE pin M1 : 0
value of MODE pin M2 : 1
value of CFG_RDY (INIT_B) : 1
DONEIN input from DONE pin : 1
IDCODE not validated while trying to write FDRI : 0
write FDRI issued before or after decrypt operation: 0
Decryptor keys not used in proper sequence : 0
INFO:iMPACT:2219 - Status register values:
INFO:iMPACT - 0011 0111 1011 1000 0000 0000 0000 0000
INFO:iMPACT:579 - '3': Completed downloading bit file to device.
INFO:iMPACT - '3': Checking done pin....done.
'3': Programmed successfully.
Elapsed time = 5 sec.
sigabort.

Done!

10.3 ANEXO 3: Traducción a VHDL de parte del código SIMD-BPD

```
-- COMUNICACION FPGA - POWERPC : ENTRADA DATOS Y SALIDA POR BUFFER
-- for --> substituir por flip-flop --> paralelo

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

-------<<<<<<<<<<<<<---------PODEMOS HACER EN PARALELO LOS KISx


entity params is
port( pattern: IN std_logic_vector(3 downto 0); ----puntero a char coge 4 bits, hacemos entrada de 4 bits
      texto: IN std_logic_vector(3 downto 0);   ----puntero a char coge 4 bits, hacemos entrada de 4 bits
      k: IN integer; ----numero de errores posibles
      count: OUT integer; ---- resultado
)
end params;

architecture funciones of params is
      signal res: integer;
      signal ptt: std_logic_vector(3 downto 0);
      signal txt: std_logic_vector(3 downto 0);
      signal num_err: integer;
begin
      process(pattern,texto,k) --cuando acaba el proceso, se cambia el valor de los signals, no al asignarlas;
                               --en cambio, las variables cambian en el acto
            variable l_pattern: integer;
            variable l_text: integer;
      begin
            function searchAutom_kis1 ( ) return unsigned integer is
            variable count: unsigned integer;
            type m128 is array(127 downto 0) of bit; --- como trabajamos con 128 bits, usamos un
                                                     ---array para poder trabajar con bits
                                                     ---bit_vector(127 downto 0)
            variable mask1: m128;
            variable mask2: m128;
            variable mask3: m128;
            variable mask4: m128;
            variable mask5: m128;
            type MYVECTOR is array(255 downto 0) of m128; --> B[sigma] -- __m128i B[sigma]
            variable B: MYVECTOR;
            variable BB: MYVECTOR;
            variable D: m128;
            variable x: m128;
            variable found_var: integer;
            variable found_var2: integer;
            variable i: unsigned integer;
            variable rest: unsigned integer; ------NATURAL
            variable bv: m128;
            variable cv: m128;
            variable c: unsigned;
            variable j: unsigned;
```

```
variable m: unsigned;
variable pos: unsigned;
variable index: unsigned;
m:=l_pattern;

constant BASE_SIGMA: integer := 0;

begin
for c in sigma-1 downto 0 loop
        B(c)<=shift_bit_masks(9); --> shift_bit_masks(9)=_mm_set1_epi32(ones)
                                  -- sets the 4 signed 32-bit integer values to "ones"
                                  -- (r0,r1,r2,r3) 4x8=32 bits
end loop;
for j in l_pattern-1 downto 0 loop
        index:= (pattern(j) - BASE_SIGMA);
        mm_slli_bit_specialized(shift_bit_masks(8),j,x);
        mask2:=_mm_sub_epi32(shift_bit_masks(9),x);
        --> traducir _mm_sub_epi32 (shift_bit_masks(9),x)
        B(index)<=_mm_and_si128(B(index),mask2);
        --> traducir _mm_and_si128(B(index),mask2)
end loop;

mm_slli_bit_1(mask3, shift_bit_masks(9));
--> traducir mm_slli_bit_1(mask3, shift_bit_masks(9))

for c in sigma downto 0 loop
        BB(c)<=shift_bit_masks(0); --> traducir shift_bit_masks(0)

        for j in m-2 downto 0
                mm_slli_bit_3(BB(c),BB(c)); --> traducir mm_slli_bit_3(BB(c),BB(c))
                mm_srli_bit_specialized(B(c),j,mask1);
                --> traducir mm_slli_bit_3(B(c),j,mask1)
                mm_slli_bit_1(mask2,mask3); --> traducir mm_slli_bit_1(mask2,mask3)
                mask2:=_mm_xor_si128(mask2,shift_bit_masks(9));
                --> traducir _mm_xor_si128(mask2,shift_bit_masks(9))
                mask2:=_mm_and_si128(mask1,mask2);
                --> traducir _mm_and_si128(mask1,mask2)
                BB(c)<=_mm_add_epi64(BB(c),mask2);
                --> traducir _mm_add_epi64(BB(c),mask2)
        end loop;

        mask1:=shift_bit_masks(0); --> traducir shift_bit_masks(0)
        mask2:=shift_bit_masks(0); --> shift_bit_masks(0);
        mm_slli_bit_1(mask4, shift_bit_masks(9));
        --> traducir mm_slli_bit_1(mask4, shift_bit_masks(9))
        mm_slli_bit_1(mask4,mask4); --> mm_slli_bit_1(mask4,mask4);
        mask4:=_mm_xor_si128(mask4,shift_bit_masks(9));
        --> _mm_xor_si128(mask4,shift_bit_masks(9));
        x:=shift_bit_masks(8); --> traducir shift_bit_masks(8)

        for j in m-2 downto 0 loop
                mm_slli_bit_3(mask1,mask1); --> mm_slli_bit_3(mask1,mask1)
                mask1:=_mm_add_epi64(mask1,x); --> _mm_add_epi64(mask1,x)
                mm_slli_bit_3(mask3,mask3); --> mm_slli_bit_3(mask3,mask3)
                mask3:=_mm_add_epi64(mask3,mask4); --> _mm_add_epi64(mask3,mask4)
        end loop;
```

```
                mm_srli_bit_3(mask2,mask1); --> mm_srli_bit_3(mask2,mask1)
                mm_slli_bit_3(mask2,mask2); --> mm_slli_bit_3(mask2,mask2)
                mask2:=_mm_add_epi64(mask2,mask4); --> _mm_add_epi64(mask2,mask4)
                mask5:=mask4;
                mask4:=shift_bit_masks(8); --> shift_bit_masks(8)
                mm_slli_bit_1(mask4,mask4); --> mm_slli_bit_1(mask4,mask4)
                D:=mask3;
                count:=0;
                --type m128 is array(127 downto 0) of bit;
                variable core1: m128;
                variable core2: m128;
                variable core3: m128;
                variable core4: m128;
                found_var2:=_mm_cvtsi128_si32(mask4); -->  _mm_cvtsi128_si32(mask4)

                for pos in l_text-1 downto 0 to loop
                        index:=texto(pos)-BASE_SIGMA;
                        mm_srli_bit_3(x,D); --> mm_srli_bit_3(x,D)
                        x:=_mm_or_si128(x,BB(index)); --> _mm_or_si128(x, BB(index))
                        mm_slli_bit_1(core_1,D); --> mm_slli_bit_1(core_1,D)
                        core_1:=_mm_or_si128(core_1,mask1); --> _mm_or_si128(core_1,mask1)
                        mm_slli_bit_4(core_2,D); --> mm_slli_bit_4(core_2,D)
                        core_2:=_mm_or_si128(core_2,mask2); --> _mm_or_si128(core_2,mask2)
                        core_3:=_mm_add_epi64(x,mask1); --> _mm_add_epi64(x,mask1);
                        core_3:=_mm_xor_si128(core_3,x); --> _mm_xor_si128(core_3,x);
                        mm_srli_bit_1(core_4,core_3); --> mm_srli_bit_1(core_4,core_3);
                        core_4:=_mm_and_si128(core_4,mask3); --> _mm_and_si128(core_4,mask3);
                        D:=_mm_and_si128(core_1,core_2); --> _mm_and_si128(core_1,core_2);
                        D:=_mm_and_si128(D,core_4); --> _mm_and_si128(D,core_4);
                        found_var:=_mm_cvtsi128_si32(D); --> _mm_cvtsi128_si32(D);

                        if not(found_var and found_var2) then
                                D:=_mm_or_si128(D,mask5); -->  _mm_or_si128(D,mask5)
                                count:=count+1;
                                return 0;
                        end if;
                end loop;
        end loop;
        return count;
end searchAutom_kis1;
```

11. Documentación

1. http://1.http://es.wikipedia.org/wiki/Tecnología_de_la_información
2. http://dewey.uab.es/pmarques/tic.htm
3. http://fpgalibre.sourceforge.net/#tp2
4. http://en.wikipedia.org/wiki/Field-programmable_gate_array
5. http://en.wikipedia.org/wiki/Flip-flop_%28electronics%29
6. http://en.wikipedia.org/wiki/Breadboard
7. http://en.wikipedia.org/wiki/Fan-in
8. http://en.wikipedia.org/wiki/Combinational_logic
9. http://en.wikipedia.org/wiki/Routing
10. http://en.wikipedia.org/wiki/Network
11. http://en.wikipedia.org/wiki/Programmable_logic_device
12. http://en.wikipedia.org/wiki/Logic_gate
13. http://en.wikipedia.org/wiki/Application-specific_integrated_circuit
14. http://en.wikipedia.org/wiki/Time_to_market
15. http://en.wikipedia.org/wiki/Non-recurring_engineering
16. http://en.wikipedia.org/wiki/Complex_programmable_logic_device
17. http://www.irb.hr/en/cir/education/courses/fpga/FPGA/fpga_sklopovi/
18. http://www.alegsa.com.ar/Dic/routing.php
19. http://diccionario.reverso.net/ingles-sinonimos/
20. http://translate.google.com/translate_t?langpair=enles
21. http://www.irb.hr/en/cir/education/courses/fpga/FPGA/fpga_sklopovi/fpga.jpg
22. http://en.wikipedia.org/wiki/Lookup_table

23. http://en.wikipedia.org/wiki/Fanout

24. http://klabs.org/richcontent/Tutorial/MiniCourses/architecture_logic_mapld2001/Architecture_Section/04_FPGA_Architecture.PDF

25. http://www.xilinx.com/company/press/kits/asmbl/asmbl_arch_pres.pdf

26. http://www.pldesignline.com/;jsessionid=TH2KCBMOGDDPAQSNDLRCKH0CJUNN2JVN

27. http://www.pldesignline.com/showArticle.jhtml?articleID=197005332

28. http://es.wikipedia.org/wiki/SRAM

29. http://en.wikipedia.org/wiki/Digital_signal_processor_DSP

30. http://en.wikipedia.org/wiki/Software-defined_radio

31. http://en.wikipedia.org/wiki/System-on-a-chip

32. http://en.wikipedia.org/wiki/Brute-force_attack

33. http://en.wikipedia.org/wiki/High_Performance_Computing

34. http://en.wikipedia.org/wiki/FFT

35. http://en.wikipedia.org/wiki/Convolution

36. http://en.wikipedia.org/wiki/Reconfigurable_computing

37. http://www.xilinx.com/company/gettingstarted/fpgavsasic.htm

38. http://en.wikipedia.org/wiki/Floorplanning

39. http://en.wikipedia.org/wiki/Electronic_design_automation

40. http://en.wikipedia.org/wiki/Turnaround

41. http://www.soccentral.com/results.asp?CategoryID=488&EntryID=15887

42. http://www.controleng.com/index.asp?layout=articleCtl&articleid=CA607224

43. http://www.irb.hr/en/cir/education/courses/fpga/FPGA/fpga_sklopovi/comparison.jpg

44. http://www.octera.com/tips_and_techniques/fpga_vs_asic.php4

45. http://es.wikipedia.org/wiki/Expresi%C3%B3n_regular

46. http://en.wikipedia.org/wiki/SNOBOL

47. http://es.wikipedia.org/wiki/Snobol

48. http://en.wikipedia.org/wiki/Pattern_matching#Pattern_matching_and_strings

49. http://en.wikipedia.org/wiki/Pattern_matching

50. http://en.wikipedia.org/wiki/Agrep

51. http://en.wikipedia.org/wiki/Fuzzy_string_searching

52. http://en.wikipedia.org/wiki/Grep

53. http://www.cs.sunysb.edu/~algorith/implement/agrep/implement.shtml

54. http://www.cs.sunysb.edu/~algorith/files/approximate-pattern-matching.shtml

55. http://www.cs.sunysb.edu/~algorith/files/string-matching.shtml

56. http://books.google.es/books?
id=GTTPGJctsMAC&printsec=frontcover&dq=flexible+pattern+matching&ei=
hDMWSIvsKouOywTxwKWCCA&sig=KwM6jMbTz_NJpiSJW-
hUJdaYT8Q#PPT1,M1

57. http://morguapu.upc.es/crom/course/view.php?
id=5&page=Historia+de+la+Microinformatica&gid=0&uid=0

58. http://morguapu.upc.es/crom/course/view.php?
id=5&page=Medi+Ambient+i+Residus+inform%C3%A0tics&gid=0&uid=0

59. http://www.nist.gov/dads/HTML/nondetermFiniteStateMach.html

60. http://www.xilinx.com/univ/

61. http://ieeexplore.ieee.org/

62. http://en.wikipedia.org/wiki/Nondeterministic_finite_state_machine

63. http://forums.xilinx.com/xlnx/board/message?board.id=EDK&message.id=702

64. http://www.xilinx.com/products/devkits/XUPV2P.htm

65. http://git.zerfleddert.de/cgi-bin/gitweb.cgi/usb-driver?a=blob_plain;f=README;hb=HEAD

66. http://www.fpgacentral.com/fpga-product/digilent-inc/nexys

67. http://stefan.endrullis.de/en/xilinx_ise_8.1.html#impact

68. VHDL para Síntesis I. Introducción al Lenguaje VHDL; KOLDO BASTERRETXEA (documentación digital)

69. Lenguaje VHDL. Dispositivos Digitales; UNIVERSIDAD DE ZARAGOZA, IEC (documentación digital)

70. Descripción en VHDL de circuitos aritméticos para campos finitos GF(2^m); MARIO ALBERTO GARCÍA-MARTÍNEZ (documentación digital)

71. C/UNIX Function for VHDL Testbenches (with additional notes on Unix pipes & rsh) 2002 - Update 2004; MICHAEL J.KNIESER, FRANCIS G.WOLFF, CHRIS A.PAPACHRISTOU (documentación digital)

72. Official ISE 9.1i Quick Start Tutorial; XILINX 2007 (documentación digital)

73. El Lenguaje VHDL: objetos, tipos de datos, operadores y subprogramas; UNIVERSIDAD DE ALCALÁ, ITT-SISTEMAS ELECTRÓNICOS (documentación digital)

74. VHDL Math Tricks of the Trade; JIM LEWIS (Director of Training, SynthWorks Design Inc) 2003 (documentación digital)

75. SIMD Math Library Specification for CELL Broadband Engine Architecture, Version 1.1., CBEA JSRE Series; IBM 2007

76. Introducción a la Programación en VHDL; Dpto. Arquitectura de Computadores y Automática - UNIVERSIDAD COMPLUTENSE DE MADRID, MARCO SÁNCHEZ-ÉLEZ, MARIA CARMEN MOLINA
77. Increased Bit-Parallelism for Approximate and Multiple String Matching; HEIKKI HYYRÖ, PRESTO, KIMMO FREDRIKSSON and GONZALO NAVARRO
78. Practical Methods for Approximate String Matching; HEIKKI HYYRÖ
79. Flexible Pattern Matching in Strings. Practical on-line search algorithms for texts and biological sequences; GONZALO NAVARRO and MATHIEU RAFFINOT
80. Design of Efficient FPGA Circuits for Matching Complex Patterns in Network Intrusion Detection Systems; CHRISTOPHER R. CLARK, Georgia Institute of Technology, 2003
81. Increased Bit-Parallelism for Approximate and Multiple String Matching
82. VHDL. Lenguaje para síntesis y modelado de circuitos. 2ª edición; FERNANDO PARDO, JOSÉ A.BOLUDA; Ediciones Ra-Ma
83. http://arstechnica.com/articles/paedia/cpu/simd.ars
84. http://en.wikipedia.org/wiki/SIMD
85. http://svenand.blogdrive.com/archive/40.html
86. http://www.fpga-faq.com/FAQ_Pages/0044_Xilinx_Parallel_on_Linux.htm
87. http://git.zerfleddert.de/cgi-bin/gitweb.cgi/usb-driver?a=blob_plain;f=README;hb=HEAD
88. http://gentoo-wiki.com/HOWTO_Xilinx#Installing_Cable_Driver_Kernel_Modules

89. http://www.itee.uq.edu.au/~listarch/microblaze-uclinux/archive/2007/03/msg00101.html

90. http://stefan.endrullis.de/en/xilinx_ise_8.1.html#impact

91. http://www.seas.upenn.edu/~cse372/tutorial/

92. http://virtex2pro.blogspot.com/2008/02/integrating-vhdl-design-into-peripheral.html

93. http://www.digilent.cc/Products/Detail.cfm?Nav1=Products&Nav2=Programmable&Prod=XUPV2P

94. Xilinx Univerity Program Virtex-II Pro and FPGA Guide. Reference Manual; XILINX 2005. Revision 2007

95. Fundamentos de lógica digital con diseño VHDL. 2ª edición; STEPHEN BROWN, ZVONKO VRANESIC; Ediciones McGrawHill

96. http://www.xilinx.com/support/documentation/system_ace_solutions.htm

97. http://avatar.ecen.okstate.edu/wiki/index.php/Accessing_files_from_a_CF_card_using_SysAce

98. Hard- and Software Modules for Streaming Video Capture and Compression Running on a Distributed FPGA Platform Using uClinux; IGOR QIHUA SHANG

99. http://asic-soc.blogspot.com/2007/11/asynchronous-fifo-design.html

100. http://forums.xilinx.com/xlnx/board/message?board.id=EMBEDDED&message.id=409

101. http://www.quicknet.se/hdc/hdl/educaton/inout/index.htm

102. http://www.ece.sunysb.edu/~jochoa/vhd_writefile_tutorial.htm

103. http://www.ece.neu.edu/info/vhdl/Unit2_Part4/Unit2_Part4.html

104. http://www.semdesigns.com/products/formatters/VHDLFormatterExample.html

105. http://www.sccs.swarthmore.edu/users/03/azuppann/cs23-lab1/index.html

106. http://www.mydatabasesupport.com/forums/arch/254276-problems-using-xilfatfs-xup-v2pro-board.html

107. http://www.cs.washington.edu/research/lis/empart/xup_ppc_linux.shtml

108. http://www.fpga4fun.com/

109. http://svenand.blogdrive.com/archive/41.html

110. http://www.eece.unm.edu/xup/ml300ppc405.htm

111. http://www.xilinx.com/support/techsup/tutorials/tutorials8.htm

112. http://avatar.ecen.okstate.edu/wiki/index.php/Creating_Peripherals

113. http://avatar.ecen.okstate.edu/wiki/index.php/I/O

114. http://www.bioinspired.com/users/ajg112/electronics/edkHOWTO1.shtml

115. http://virtex2pro.blogspot.com/2008/02/integrating-vhdl-design-into-peripheral.html

116. http://www.wipo.int/pctdb/en/wo.jsp?wo=2007130818&IA=WO2007130818&DISPLAY=CLAIMS

117. http://avatar.ecen.okstate.edu/wiki/index.php/Hello_World!_and_Ace_Files

118. http://forums.xilinx.com/xlnx/board/message?board.id=Virtex&thread.id=635

119. http://www.fpga-faq.com/archives/87175.html

120. http://www.fpgacentral.com/forums/fpga-devices/xilinx-devices-0

121. http://www.edaboard.com/ftopic313843.html

122. http://www.vlsibank.com/titles.asp?cat=FPGA

123. http://forums.opalkelly.com/forumdisplay.php?f=28

124. http://www.velocityreviews.com/forums/f5-hardware.html

125. http://stefan.endrullis.de/en/xilinx_ise_8.1.html#impact

126. http://chaos.ece.mcgill.ca/?q=XilinxUSBGutsy

127. http://groups.google.com/group/comp.arch.fpga/msg/2dfa36541174a4f2

12. Herramientas Utilizadas

1. http://translate.google.com/translate_t?langpair=enles
2. http://diccionario.reverso.net/ingles-sinonimos/
3. http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm
4. http://www.grec.net/home/cel/dicc.htm
5. http://www.softcatala.org/forum/llengua/messages/74.html
6. http://www.softcatala.org/traductor/
7. http://www.digilentinc.com/Products/Detail.cfm?Prod=NEXYS&Nav1=Products&Nav2=Programmable
8. http://www.google.com/codesearch
9. https://registrationcenter.intel.com/RegCenter/RegisterSNInfo.aspx?sn=NZCR-GB3JL6TW&EmailID=ariadna.trueba@gmail.com&Sequence=738095