Técnicas de ataque por fuerza bruta contra MD5 mediante FPGA

Alvaro Gamez
alvaro.gamez@hazent.com

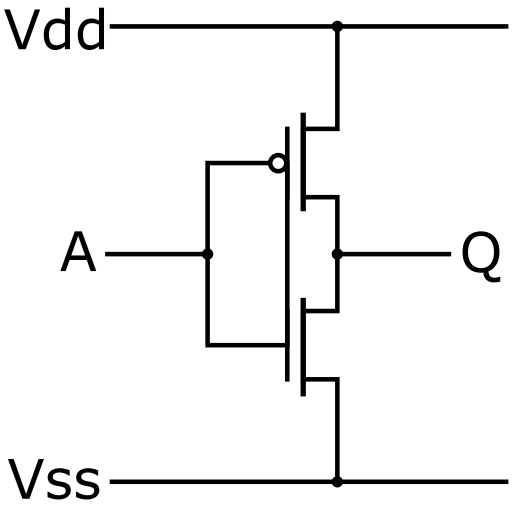
4 de febrero de 2012

Contenidos

1. Electrónica digital básica
   - Circuitos elementales
   - Circuitos complejos: procesadores
2. Circuitos digitales reconfigurables
   - FPGAs
   - VHDL
3. Implementación de MD5 en VHDL
   - Algoritmo MD5: explicación
   - Implementación del algoritmo MD5 en VHDL
   - Conclusiones y resultados

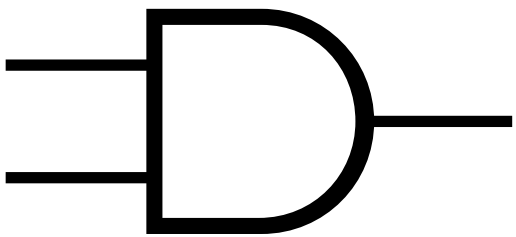
¿Qué hay dentro de un circuito integrado? (I)

Transistores y puertas lógicas



- Una puerta lógica está formada por varios transistores MOS complementarios (un tipo P y un tipo N).
- Puede tener una o más entradas, pero sólo tiene una salida.

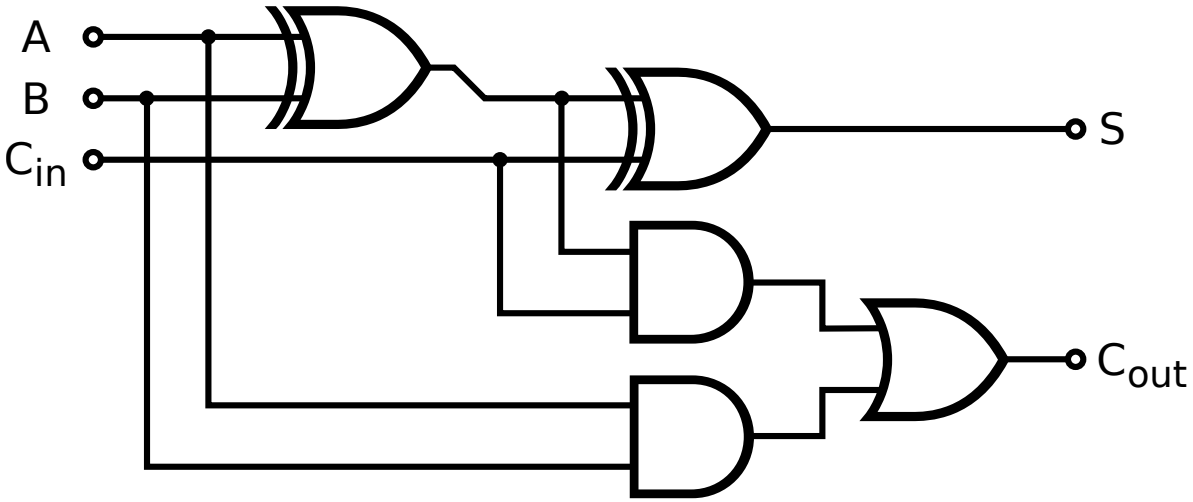
Esta otra puerta, representada por su símbolo, es una puerta AND.



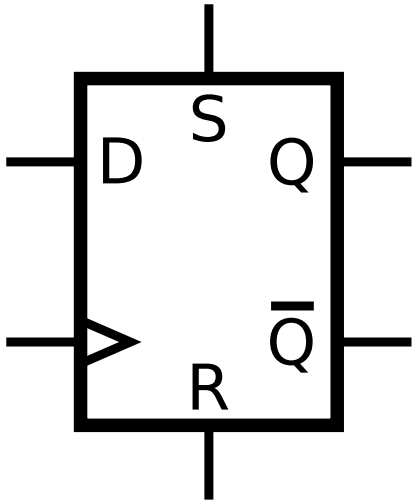
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

¿Qué hay dentro de un circuito integrado? (II)

Circuitos combinacionales y registros



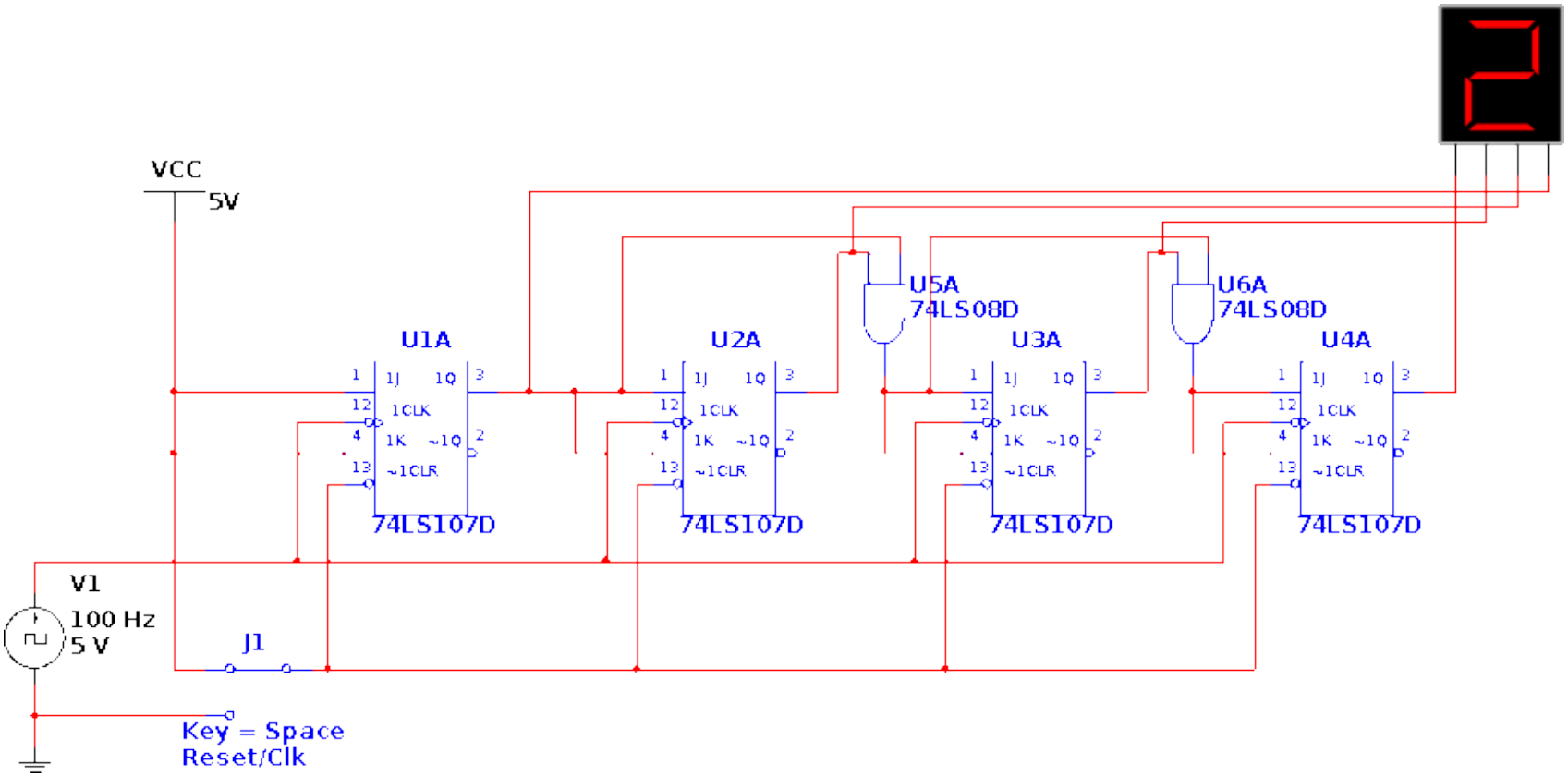
Sumador completo (con acarreo): $S = A + B + C_{in}$. Su salida depende únicamente de sus entradas.



Registro tipo D: unidad básica de almacenamiento Su salida depende únicamente de sus entradas, pero solo cambia con un ciclo de reloj.

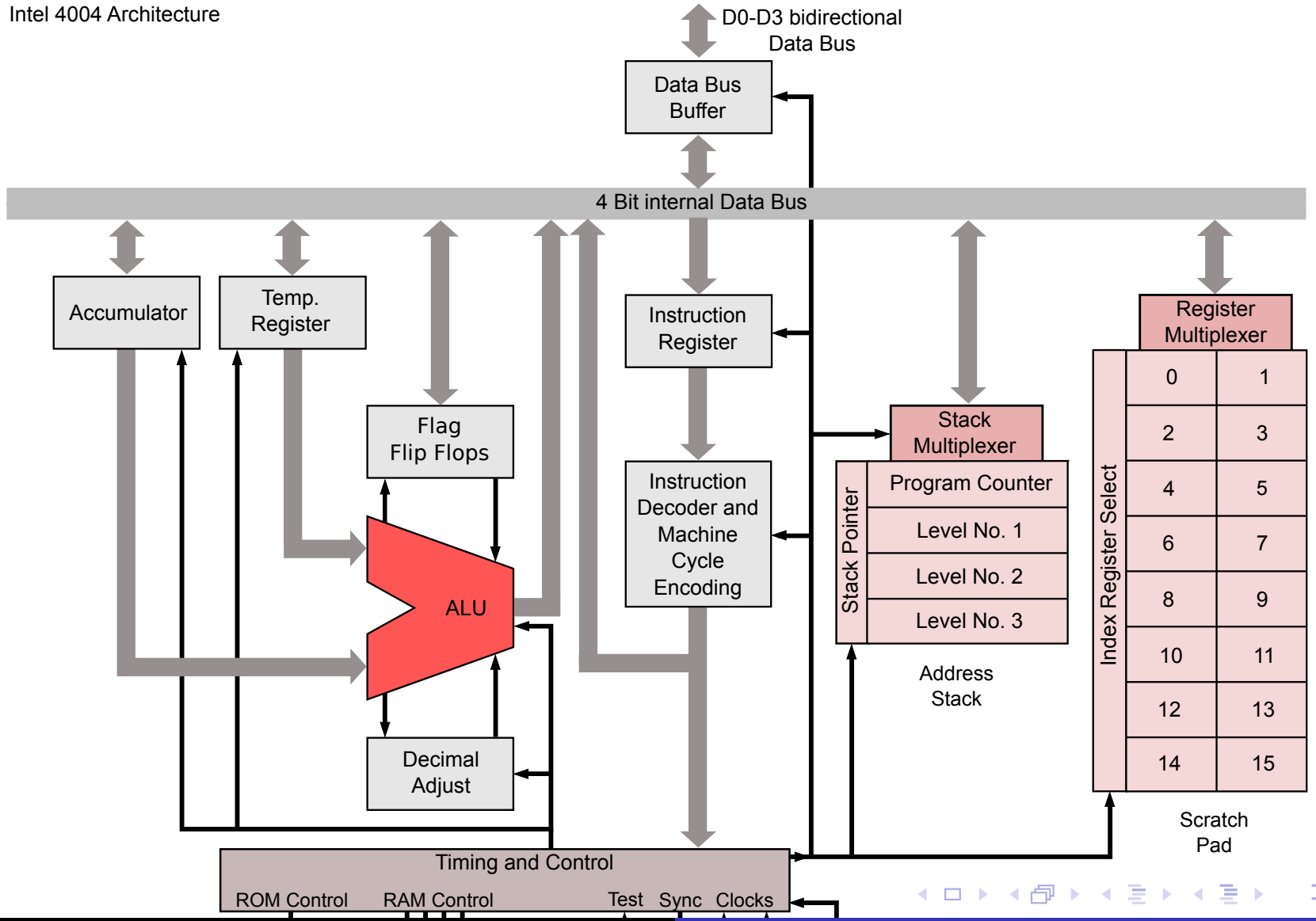
¿Qué hay dentro de un circuito integrado? (II)

Máquinas de estados



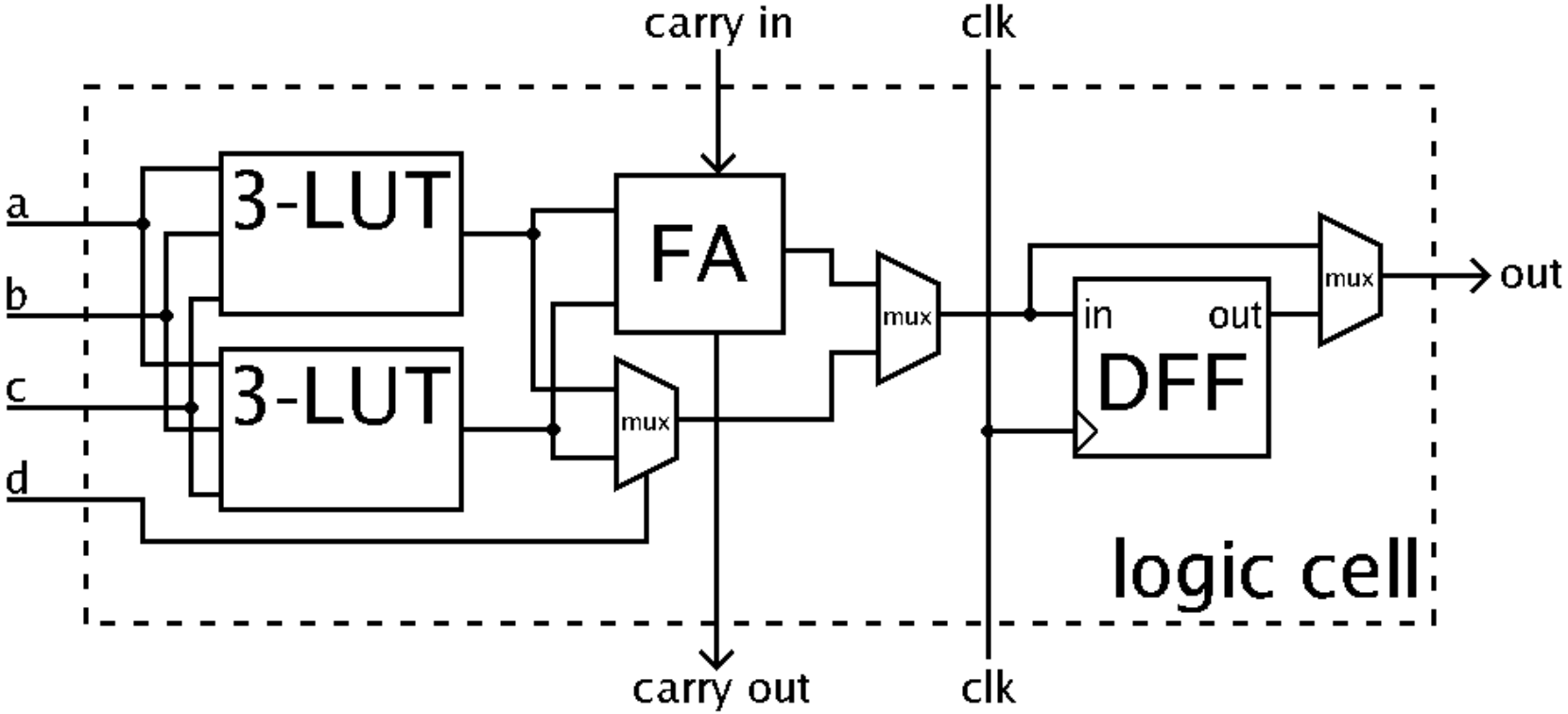
Contador de 4 bits: desde “0000” (0) hasta “1111” (15).
Su salida depende de sus entradas y de su estado actual, y su estado cambia únicamente con un ciclo de reloj.

¿Qué hay dentro de un procesador?



¿Qué hay dentro de una FPGA?

En el interior de una FPGA hay una enorme matriz de circuitos interconectados entre sí. Cada una de las celdas de esta matriz (en el orden de decenas de miles) está formada por diversos circuitos digitales como los que hemos visto anteriormente.



¿Cómo se configuran las FPGAs?

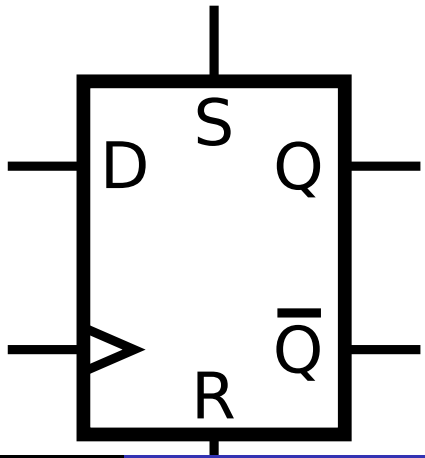
Lenguajes de descripción hardware: VHDL

El código VHDL...

... no es un programa.
... no se ejecuta.

```
1  DFF : process(RST, CLK)
2  begin
3    if RST = '1' then
4      Q <= '0';
5    elsif rising_edge(CLK) then
6      Q <= D;
7      Q <= not D;
8    end if;
9  end process DFF;
```

El código VHDL... **define qué hardware utilizar y cómo interconectarlo.**



¿Cómo se configuran las FPGAs?

Lenguajes de descripción hardware: VHDL

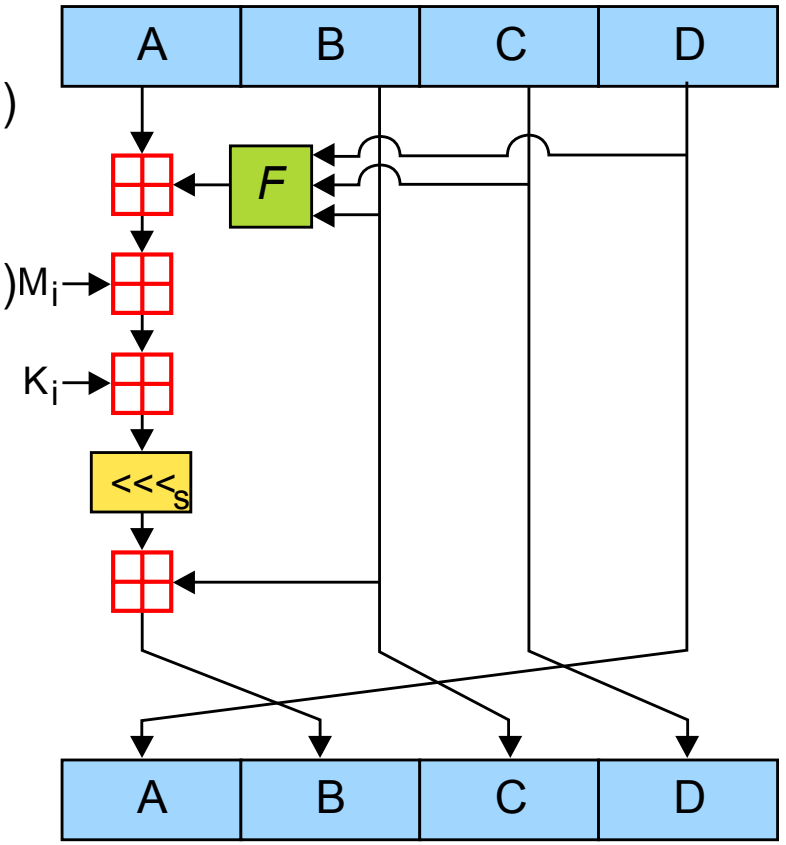
```
1  -- import std_logic from the IEEE library
2  library IEEE;
3  use IEEE.std_logic_1164.all;
4
5  -- this is the entity
6  entity ANDGATE is
7    port (
8      I1 : in std_logic;
9      I2 : in std_logic;
10     O  : out std_logic);
11 end entity ANDGATE;
12
13 architecture RTL of ANDGATE is
14 begin
15   O <= I1 and I2;
16 end architecture RTL;
```

Algoritmo MD5 en pseudocódigo (I)

```
1  //Process the message in successive 512-bit chunks:
2  for each 512-bit chunk of message
3      break chunk into sixteen 32-bit little-endian words w[j]
4      //Initialize hash value for this chunk:
5      var int a := h0
6      var int b := h1
7      var int c := h2
8      var int d := h3
9      //Main loop:
10     main_loop_on_next_slide()
11     //Add this chunk's hash to result so far:
12     h0 := h0 + a
13     h1 := h1 + b
14     h2 := h2 + c
15     h3 := h3 + d
16 end for
```

Algoritmo MD5 en pseudocódigo (II)

```
1  for i from 0 to 63
2   if 0 <= i <= 15 then
3    f := (b and c) or ((not b) and d)
4    g := i
5   else if 16 <= i <= 31
6    f := (d and b) or ((not d) and c)
7    g := (5*i + 1) mod 16
8   else if 32 <= i <= 47
9    f := b xor c xor d
10   g := (3*i + 5) mod 16
11  else if 48 <= i <= 63
12   f := c xor (b or (not d))
13   g := (7*i) mod 16
14  temp := d
15  d := c
16  c := b
17  b := b + (a + f + k[i] + w[g]) <<< r[i]
18  a := temp
19 end for
```



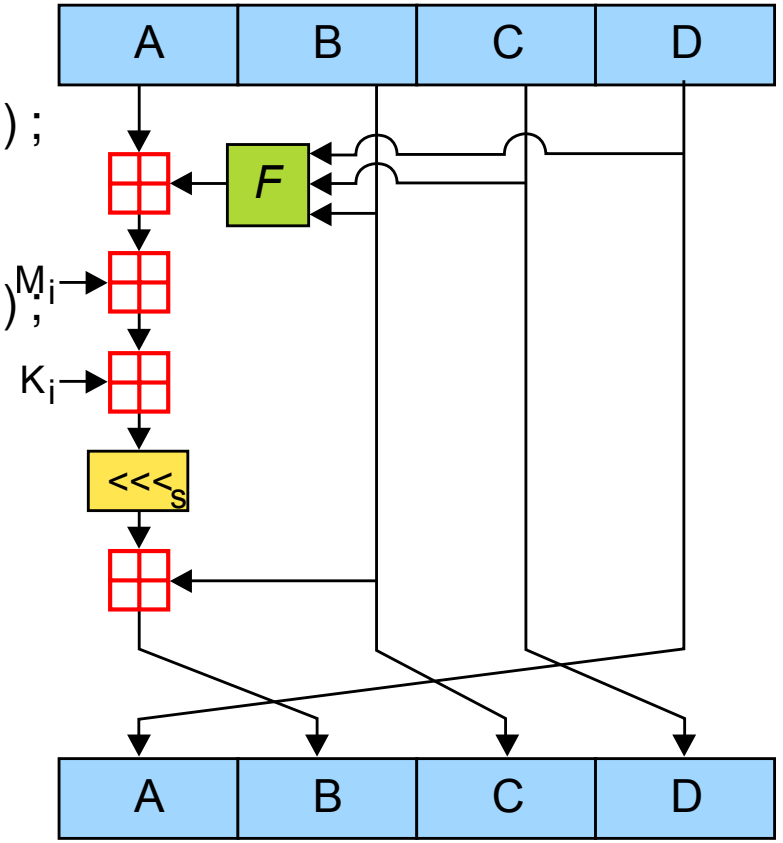
Algoritmo MD5 en VHDL (I)

En VHDL es imperativo definir un módulo en función de sus entradas y sus salidas. Nuestro módulo MD5 tiene como entradas un reloj, una señal de reset, una señal de start y 512 bits de datos distribuidos en 16 palabras de 32 bits cada una.

```
1  entity md5mainloop is
2  Port (
3   clk : in std_logic;
4   rst : in std_logic;
5
6   start : in std_logic;
7   w : in dataset(0 to 15);
8
9   o : out std_logic_vector
10          (127 downto 0);
11  ov : out std_logic
12 );
13 end md5mainloop;
```

Algoritmo MD5 en VHDL (II)

```
1  main_loop: process(i, a, b, c, d, w)
2  begin
3    if(i < 16) then
4      f <= (b and c) or ((not b) and d);
5      g <= i mod 16;
6    elsif(i<32) then
7      f <= (d and b) or ((not d) and c);
8      g <= ( (5*i) + 1 ) mod 16;
9    elsif(i<48) then
10     f <= b xor c xor d;
11     g <= ( (3*i) + 5 ) mod 16;
12   elsif(i<64) then
13     f <= c xor (b or (not d));
14     g <= ( 7*i ) mod 16;
15   end if;
16 end process main_loop;
17
18 x <= a+f+K(i mod 64) + w(g);
19 y <= rotate_left(x, R(i mod 64));
```



Este código **describe y sintetiza** una combinación de puertas lógicas.

Algoritmo MD5 en VHDL (III)

```
1  after_loop: process(clk, running, i, a, b, c, d, y)
2  begin
3    if(rising_edge(clk) and running='1') then
4      if(i<64) then
5        i <= (i+1);
6        d <= c;
7        c <= b;
8        b <= b + y;
9        a <= d;
10     else
11       i <= 0;
12       running <= '0';
13       h0 <= h0 + a;
14       h1 <= h1 + b;
15       h2 <= h2 + c;
16       h3 <= h3 + d;
17     end if;
18   end if;
19 end process after_loop;
```

Algoritmo MD5 en VHDL (IV)

```
1  control: process(clk, rst, a, b, c, d, i, y, w)
2  begin
3    if(rising_edge(clk)) then
4      ov <= '0';
5      if(rst='1') then
6        i <= 0;
7        a <= wA; h0 <= wA;
8        b <= wB; h1 <= wB;
9        c <= wC; h2 <= wC;
10       d <= wD; h3 <= wD;
11       running <= '0';
12     elsif(running='0') then
13       if(start = '1') then
14         i <= 0;
15         running <= '1';
16       end if;
17     end if;
18   end if;
19 end process control;
```

Máxima frecuencia de funcionamiento (I)

Según el dispositivo en el que implementemos el código VHDL se sintetizará un hardware u otro: algunas FPGAs tienen LUTs con más entradas, DSPs con multiplicadores, sumadores, etc.

Además, dependiendo de la tecnología de fabricación el retardo de las puertas puede ser mayor o menor. Por tanto, la máxima frecuencia de funcionamiento del código depende de la FPGA en la que se implemente.

Nuestro módulo MD5 completo puede entregar el resultado de realizar el hash a un mensaje de menos de 448 bits (56 bytes) en menos de 75 ciclos de reloj.

Máxima frecuencia de funcionamiento (II)

Resultado de implementación en diversas FPGAs:

FPGA	Máxima frecuencia	Recursos
XC3S100E (Spartan 3E)	60 MHz	$< 60\,\%$
XC6SLX9 (Spartan 6)	100MHz	$< 15\,\%$
XC5VSX95T (Virtex 5)	130 MHz	$< 1\,\%$

Benchmark de cálculo de hash MD5 con john en un Intel i7:

```
Benchmarking: FreeBSD MD5 [32/64 X2]... DONE
Raw:    14722 c/s real, 14722 c/s virtual
```

Resultados (I)

Suponiendo que aceptamos una ocupación del 90 % de cada uno de los tres dispositivos mencionados (dejando libre el otro 10 % para la lógica de control y de comunicación con el exterior: SPI, RS232, VGA...) podríamos implementar una cantidad variable de módulos MD5 en paralelo, multiplicando efectivamente la capacidad de procesado.

FPGA	Módulos	Hashes/segundo
Spartan 3E	1	800000
Spartan 6	6	8000000
Virtex 5	90	156000000
John	16	240000

Resultados (II)

Puede mejorarse la velocidad insertando etapas de pipelining.
Puede reducirse el área compartiendo lógica entre módulos.

FPGA	Módulos	Hashes/segundo
Spartan 3E	1	800000
Spartan 6	6	8000000
Virtex 5	90	156000000
John	16	240000

¿El precio de un I7? >1000 euros.
¿El precio de una Spartan 3E? <50 euros.

Agradecimientos y referencias

- Imágenes y pseudocódigo prestados de http://en.wikipedia.org
- Xilinx (principal fabricante de FPGAs): http://www.xilinx.com/
- Intel 4004: http://www.intel.com/museum/archives/4004.htm
- Digital Integrated Circuits: A Design Perspective; J.M. Rabaey, Prentice Hall, 1996

Gracias a todos por venir.
¿Preguntas?