

Laboratório de Sistemas Digitais

Como ordenar números inteiros de uma forma eficiente

Introdução

Em alguns dos projetos finais de Laboratórios de Sistemas Digitais, em particular o relacionado com a geração de chaves aleatórias para o euromilhões, seria mais simpático apresentar por ordem crescente os vários números aleatórios gerados. Apesar de ser possível utilizar um dos vários algoritmos de ordenação estudados nas unidades curriculares de programação, quando o número de números a ordenar é fixo e pequeno existem maneiras de os ordenar que são muito simples de implementar numa FPGA.

Redes de ordenação (*sorting networks*)

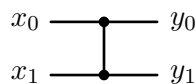
O caso mais simples ocorre quando se pretende ordenar apenas dois números. A entidade combinatória seguinte resolve facilmente esse problema:

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;

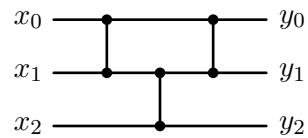
entity sort_2 is
  generic
  (
    n_bits : integer range 2 to 32
  );
  port
  (
    x0 : in  unsigned(n_bits-1 downto 0);
    x1 : in  unsigned(n_bits-1 downto 0);
    y0 : out unsigned(n_bits-1 downto 0); -- equal to min(x0,x1)
    y1 : out unsigned(n_bits-1 downto 0)  -- equal to max(x0,x1)
  );
end sort_2;

architecture combinatorial of sort_2 is
begin
  y0 <= x0 when (x0 < x1) else x1;
  y1 <= x0 when (x0 > x1) else x1;
end combinatorial;
```

Vamos representar graficamente uma instanciação desta entidade pelo desenho seguinte.

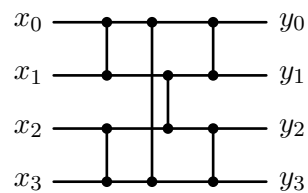


Para ordenar três números inteiros podemos simplesmente agrupar três instâncias da entidade `sort_2` da seguinte maneira (nota: entradas do lado esquerdo, saídas do lado direito):



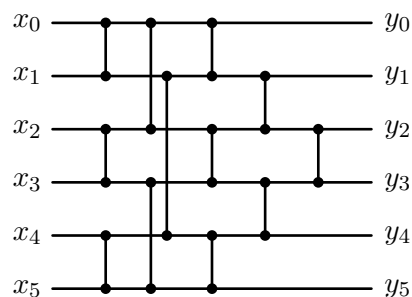
O atraso máximo de propagação da lógica combinatória corresponde ao atraso de três entidades `sort_2`.

Para ordenar quatro números inteiros podemos usar a seguinte rede de ordenação:



Note que se se eliminar x_3 e y_3 , e todas as instâncias da entidade `sort_2` que estão ligadas à linha que os une, se obtém a rede de ordenação para três números apresentada no topo da página. Mais uma vez, o atraso máximo de propagação da lógica combinatória corresponde ao atraso de três entidades `sort_2`.

Finalmente, para ordenar cinco ou seis números inteiros podemos usar a seguinte rede de ordenação (para ordenar apenas cinco, remova x_5 e y_5 , ou, em alternativa, x_0 e y_0 , e todas as instâncias da entidade `sort_2` que estão ligadas à linha que os une):



Neste caso o atraso máximo de propagação da lógica combinatória corresponde ao atraso de cinco entidades `sort_2`.

Bibliografia

Para mais informações sobre redes de ordenação de atraso mínimo consulte a seção 5.3.4 (*networks for sorting*, páginas 219 a 247) do livro

Donald E. Knuth, *Sorting and Searching*, volume 3 da *The Art of Computer Programming*. Addison-Wesley, Reading, Massachusetts, terceira edição, 1998.

Como alternativa, poderá pesquisar este assunto na internet (procure *minimum-time sorting networks*).