PJW HASH FUNCTION

STATEMENT OF THE WORK

Autores: DDLS

Última Revisão: 18 de maio de 2017

Sumário

1 Problema

Os algoritmos de hash foram uma forma que se encontrou para resumir uma certa quantidade de dados em alguns bits, existem várias formas de se desenvolver um desses algoritmos uma das mais simples é o PJW que transforma as palavras de entrada em saídas de 32 bits. Com base nessas duas implementações abaixo desenvolva um IP Core que calcule a palavra hash da mesma maneira.

Implementação 1:

```
unsigned long ElfHash ( const unsigned char *s )
{
    unsigned long   h = 0, high;
    while ( *s )
    {
        h = ( h << 4 ) + *s++;
        if ( high = h & 0xF0000000 )
            h ^= high >> 24;
        h &= ~high;
    }
    return h;
}
```

Implementação 2:

```
static unsigned int ELFHash(string str) {
    unsigned int hash = 0;
    unsigned int x = 0;
    unsigned int i = 0;
    unsigned int len = str.length();

    for (i = 0; i < len; i++)
    {
          hash = (hash << 4) + (str[i]);
          if ((x = hash & 0xF0000000) != 0)
          {
                hash ^= (x >> 24);
          }
          hash &= ~x;
    }

    return hash;
}
```

2 Interface

A interface de comunicação do seu sistema deve atender os seguintes requisitos:

- Duas entradas:
  - Valid (1 bit)
  - DataIn (32 bits)
- Duas saídas:
  - Ready (1 bit)
  - DataOut (32 bits)
- Ready será levado para 1 quando o IP estiver pronto para entrada ou saída de dados;
- Valid só será ativado quando o ready estiver levantado e servirá para a entrada de dados;
- DataIn só será gravado quando o Valid e o Ready estiverem ativos.

3 Entrada

O primeiro dado que deverá entrar é um número *n*, após isso haverá a entrada de *n* palavras *k*.

Onde 0 < n < 8

4 Saída

A saída deverá ser mostrada apenas quando o hash houver sido calculado e deve sair junto com a subida do ready após 1+n entradas de dado.

5 Caso Base

Valid	DataIn	Ready	DataOut
0	0	0	0
0	32’d2	1	0
1	32’d2	1	0
0	32’saaaa	0	0

0	32’saaaa	1	0
1	32’saaaa	1	0
0	32’sbbbb	0	0
0	32’sbbbb	1	0
1	32’sbbbb	1	0
0	0	0	0
...	...	...	...
0	0	1	32’h077788a5