NoC com criptografia SIMON

Moraes – 31/maio/2021

- **Latência worst-case para um pacote**: 140 ciclos por bloco de 128 flits. Otimizei o Simon original para levar 70 ciclos para criptografar/ descriptografar (68 *rounds*).

- O código está no link: http://bit.ly/hermes_simon
  funciona para flit de 16 e 32 bits. Só testei para buffer de profundidade 4.

- Para usar é só trocar na NoC:
  "router: entity work.RouterCC" por "router: entity work.RouterCC_S"
  Princípio: o RouterCC_S é um *wrapper* que encapsula o RouterCC e dois módulos SIMON

```
├── NOC
│   ├── Hermes_buffer.vhd
│   ├── Hermes_crossbar.vhd
│   ├── Hermes_package.vhd
│   ├── Hermes_switchcontrol.vhd
│   ├── NOC.vhd
│   ├── RouterCC.vhd
│   ├── RouterCC_S.vhd          ← novo
│   └── simon_wrapper.vhd       ← novo
├── SIMON
│   ├── simon_core.vhd              ← novo
│   └── simon_key_scheduler.vhd     ← novo
├── rcp_packet.txt
├── sim.do
├── tb.vhd
└── wave.do
```

Nenhuma alteração dos sinais do routerCC.

Recomendo fortemente usar os arquivos do projeto onde se quer inserir o Simon, pois esta versão da NoC pode ter o *crossbar* limitado, sem permitir E/S.

- Quem indica se o pacote deve ser criptografado é o bit mais significativo do primeiro flit.

- Descrição dos pacotes no *tb*:

```
constant tp : tpacket := (--start  size    src tgt  encrypt
                          ( 0,     2*FB,    0,  8,  '1'),
                          ( 0,       FB,    2,  6,  '1'),
                          ( 20,    4*FB,    0,  7,  '1'),
                          ( 12,    5*FB,    8,  0,  '1'),
                          ( 400,   20*FB,    2,  3, '1')
                         );
```

  Start é o tempo esperado para injetar o pacote, *size* é um múltiplo de FB (flits por bloco, que se traduz por: 128/TAM_FLIT).

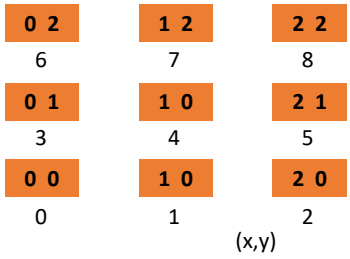
- O tb gera um relatório ***rcp_packet.txt***:

```
From: 2 To: 6 P: 82000002..00000004..00000002..00020002..00030003..00040004.. [Latency: 271]
```

```
From: 0 To: 8 P:
80000202..00000008..00000002..00020002..00030003..00040004..00050005..00060006..00070007..00080008..
[Latency: 342]

From: 8 To: 0 P:
82020000..00000014..0000000E..00020002..00030003..00040004..00050005..00060006..00070007..00080008..
00090009..000A000A..000B000B..000C000C..000D000D..000E000E..000F000F..00100010..00110011..00120012..
00130013..00140014.. [Latency: 576]

From: 0 To: 7 P:
80000102..00000010..000000A1..00020002..00030003..00040004..00050005..00060006..00070007..00080008..
00090009..000A000A..000B000B..000C000C..000D000D..000E000E..000F000F..00100010.. [Latency: 492]

From: 2 To: 3 P:
82000001..00000050..00000192..00020002..00030003..00040004..00050005..00060006..00070007..00080008..
00090009..000A000A..000B000B..000C000C..000D000D..000E000E..000F000F..00100010..00110011..00120012..
00130013..00140014..00150015..00160016..00170017..00180018..00190019..001A001A..001B001B..001C001C..
001D001D..001E001E..001F001F..00200020..00210021..00220022..00230023..00240024..00250025..00260026..
00270027..00280028..00290029..002A002A..002B002B..002C002C..002D002D..002E002E..002F002F..00300030..
00310031..00320032..00330033..00340034..00350035..00360036..00370037..00380038..00390039..003A003A..
003B003B..003C003C..003D003D..003E003E..003F003F..00400040..00410041..00420042..00430043..00440044..
00450045..00460046..00470047..00480048..00490049..004A004A..004B004B..004C004C..004D004D..004E004E..
004F004F..00500050.. [Latency: 1740]
```

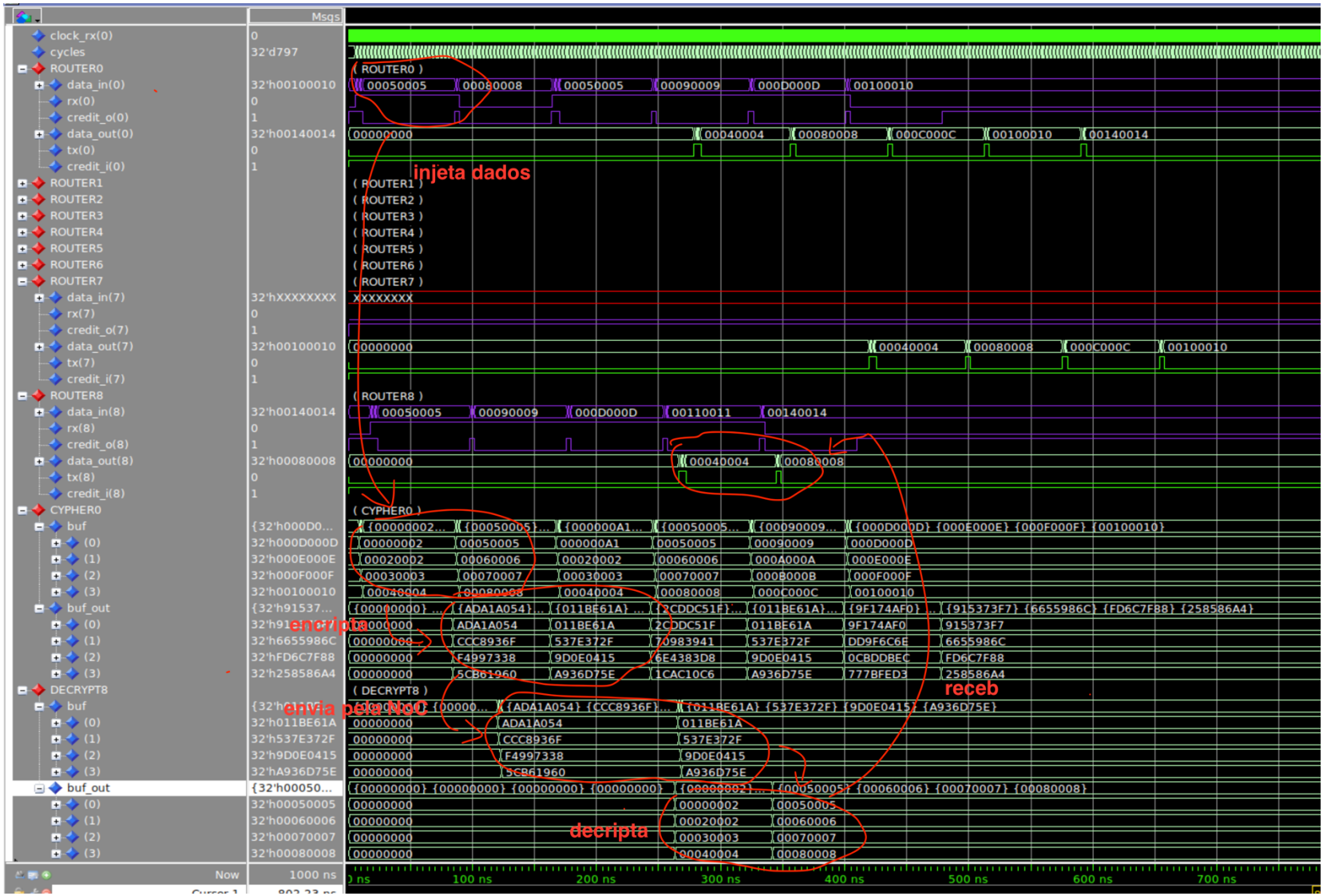
- Mesmo relatório sem criptografar (só alterando o *tb*):

```
From: 2 To: 6 P: 02000002..00000004..00000002..00020002..00030003..00040004.. [Latency: 62]

From: 0 To: 8 P:
00000202..00000008..00000002..00020002..00030003..00040004..00050005..00060006..00070007..00080008..
[Latency: 63]

From: 0 To: 7 P:
00000102..00000010..00000017..00020002..00030003..00040004..00050005..00060006..00070007..00080008..
00090009..000A000A..000B000B..000C000C..000D000D..000E000E..000F000F..00100010.. [Latency: 72]

From: 8 To: 0 P: 02020000..00000014..0000000E..00020002.. 00140014.. [Latency: 84]

From: 2 To: 3 P: 02000001..00000050..00000192..00020002..00030003..00040004..00500050.. [Latency:
183]
```



- Do 0 para o 0 temos 5 hops, logo a latência seria 25 + tam_pacote. Porém o processo nos *wrappers*, baseados em duas filas, acrescenta 11 ciclos de cada lado, mesmo sem criptografar. Esta escolha deu-se pela simplicidade de desenvolver o código.
    - Preenche uma fila com 128 flits (a profundidade depende do tamanho do flit)
    - Depois em função do MSB do *header* ou se vai para o SIMON ou se transfere os dados para a fila de saída (também com 128 flits).

Ou, seja, paga-se caro na latência, mas muito menos que em software!



Exemplo de transmissão do R0 para o R8 em uma NoC 3x3. Para ver os dados criptografados deve-se entrar no simon_wrapper dentro do router_CC_S e ver o buf e buf_out, que são os buffers de 128 que vão (ou não) para o bloco SIMON.