
Co-simulação

Leandro Augusto de Oliveira

Alexandre de Moraes Amory

{laugusto,amory}@inf.pucrs.br

Tópicos

1. Introdução

2. Ambiente de Co-simulação Desenvolvido

2.1. Estrutura Geral

2.2. Arquivo de Coordenação

2.3. Software

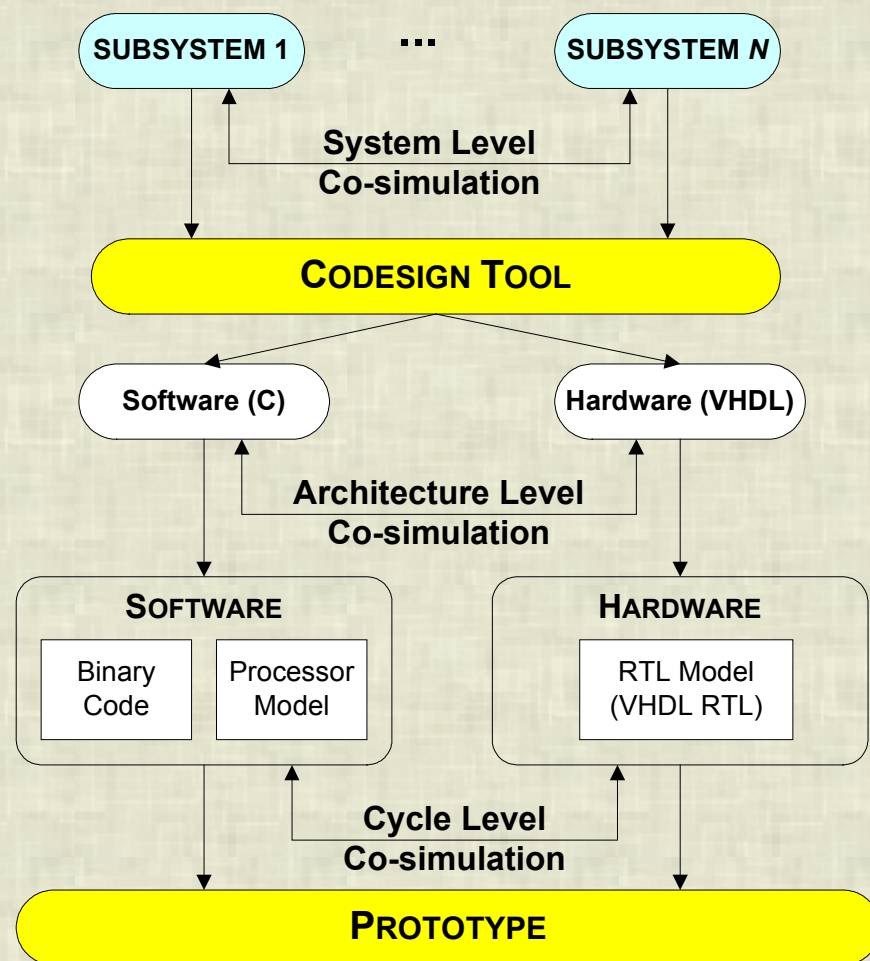
2.4. Hardware

3. Estudo de Caso

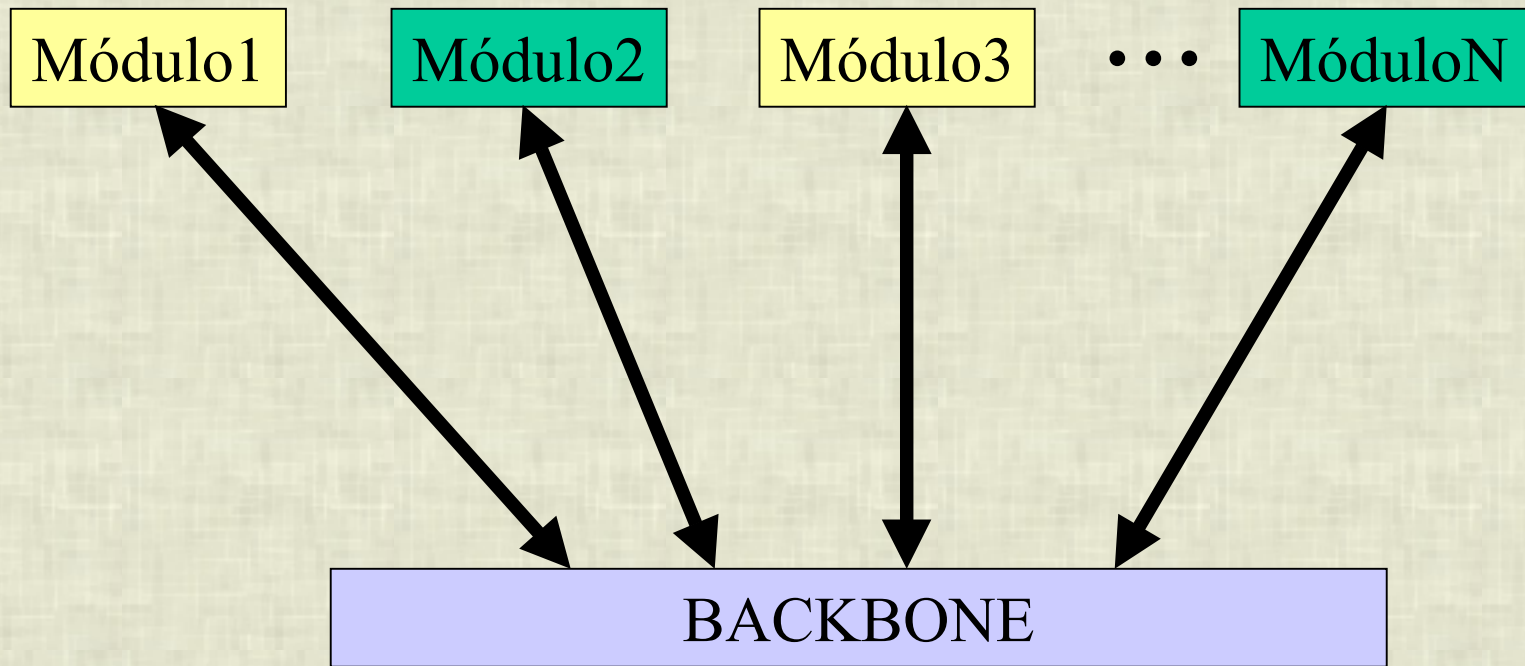
4. Conclusões

Introdução

- **Sistemas compostos por software e hardware digital**
 - hardware analógico e partes mecânicas (opcional)
- **Diferentes linguagens, MOCs e simuladores são usados**
- **É necessária uma ferramenta para coordenar a validação do sistema completo**



Organização Geral de um Ambiente de Co-simulação



software

hardware

Características de um Ambiente de Co-simulação

Validação Temporal

- algoritmo de sincronização para coordenar troca de mensagens
- uso típico: validação no nível de ciclos

Validação Funcional

- sem referencia de clock global
- validação de sistema em nível de abstração maior
- validação mais rápida que temporal

Co-simulação Geograficamente Distribuída

- descentralização de projeto
- facilita cooperação entre equipes
- gerência de IP
- gerência de licença de IP
- compartilhamento de recursos

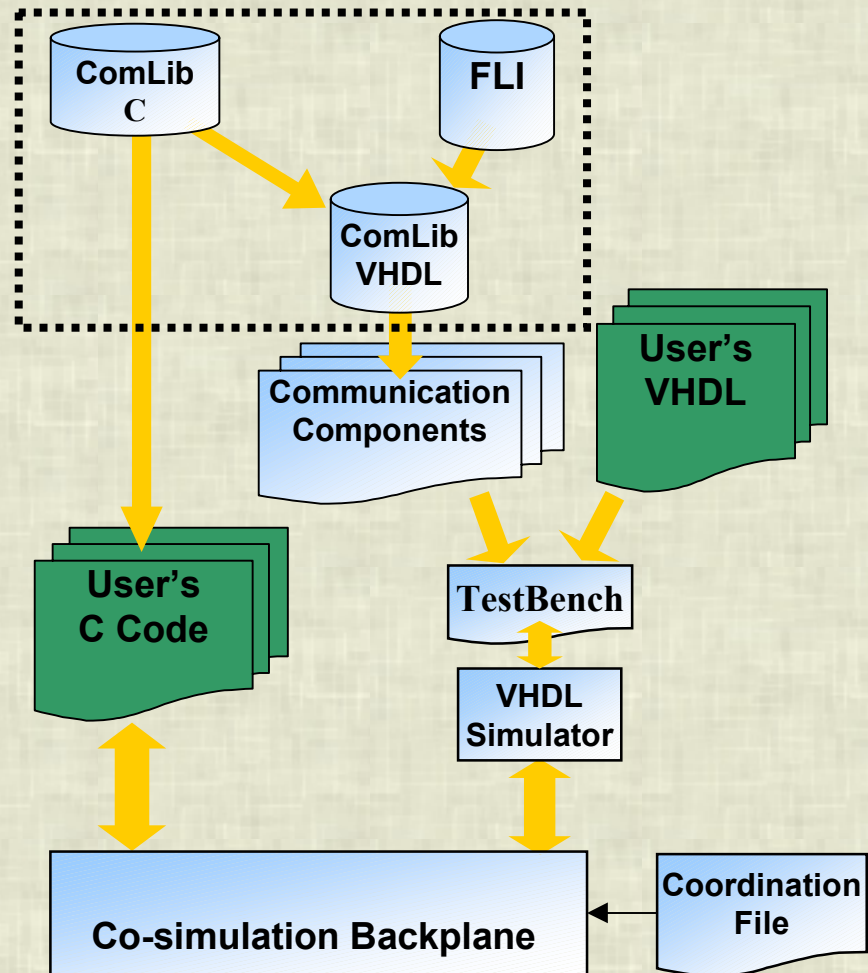
Desempenho dependente da rede !!!

Tópicos

1. Introdução
2. Ambiente de Co-simulação Desenvolvido
 - 2.1. Estrutura Geral
 - 2.2. Arquivo de Coordenação
 - 2.3. Software
 - 2.4. Hardware
3. Estudo de Caso
4. Conclusões

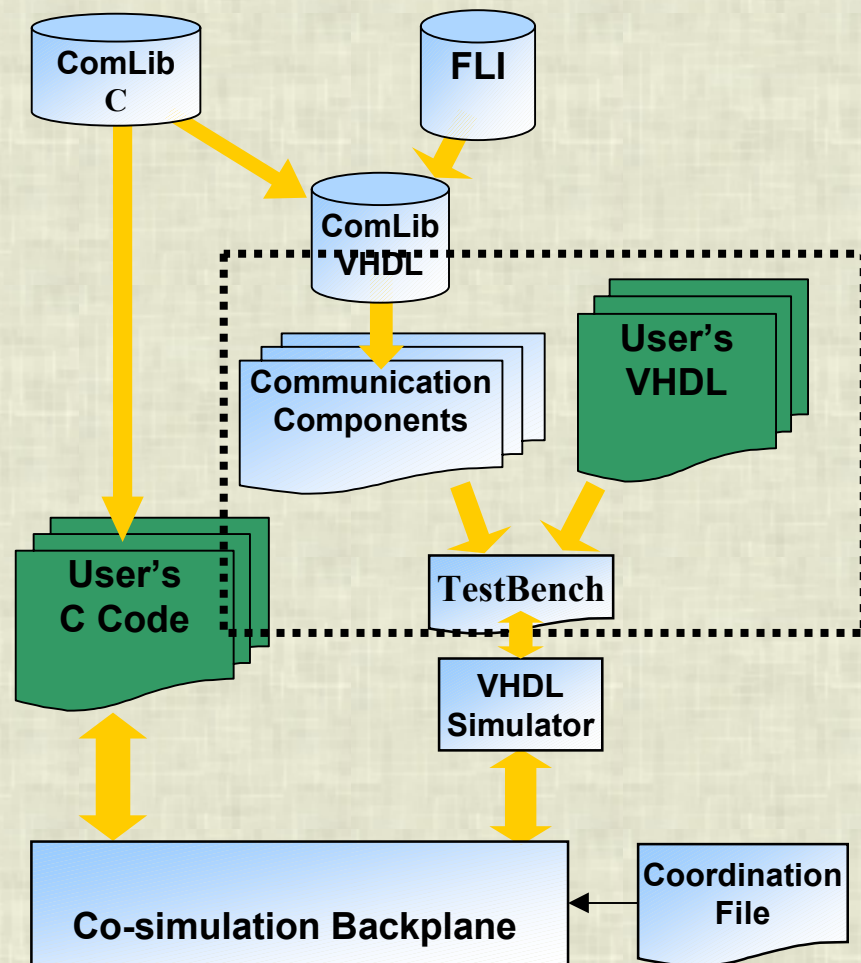
Estrutura Geral

- **ComLibC – socket**
 - csInitialize()
 - csWaitControl()
 - csSend()
 - csSendControl()
 - csIsSimulating()
 - csReceive()
- **FLI**
 - Interface entre C e VHDL
- **ComLib VHDL**
 - componentes da ComLibC
- **Data conversion**
 - realizado pela bib. de comunicação
 - Ex.: std_logic_vector to integer e vice-versa



Estrutura Geral

- **Componente de Comunicação**
 - instanciado para cada porta de I/O
- **Descrição VHDL do usuário**
 - hardware do sistema
- **Test Bench**
 - integra componentes de de comunicação com descrição do usuário



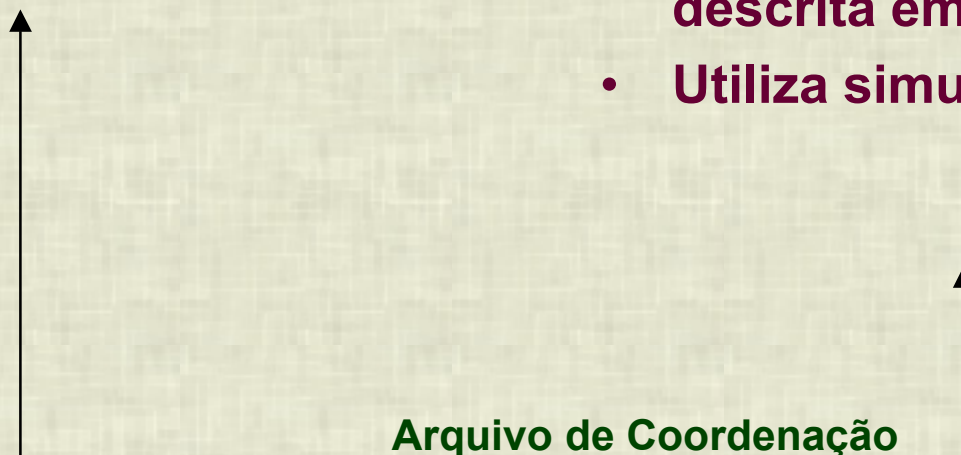
Interface Software e Hardware

Software

- o software é um programa C/C++
- compilador GCC

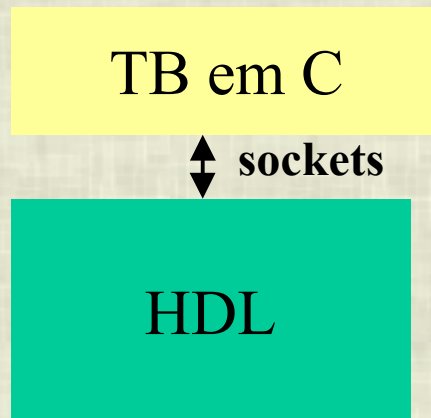
Hardware

- simulador VHDL com bib. FLI
- Com FLI uma arquitetura ou função pode ser descrita em C
- Utiliza simulador modelsim



Possíveis uso da Ferramenta

- **Co-simulação hw/sw**
- **Avaliação de partições**
 - conta o número de mensagens por cada módulo
- **Validação funcional de hardware(HDL)**



Tópicos

1. Introdução
2. Ambiente de Co-simulação Desenvolvido
 - 2.1. Estrutura Geral
 - 2.2. Arquivo de Coordenação
 - 2.3. Software
 - 2.4. Hardware
3. Estudo de Caso
4. Conclusões

Arquivo de Coordenação_(1/2)

```
module VHDL {  
  language("VHDL");  
  command("vsim Cossim_tb TB_ARCHITECTURE -do script.do");  
  machine(name="hios.inf.pucrs.br",alias="hios");  
  port {  
    int A : in;  
    int B : in;  
    control comeca : in;  
    control res: out;  
    int result : out;  }}  

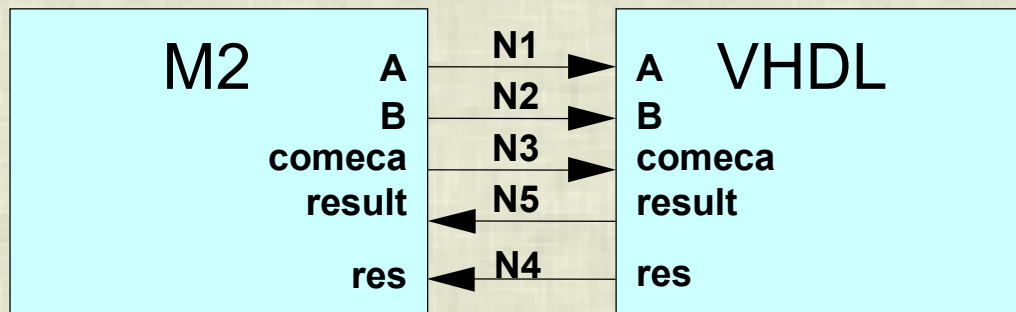
```

```
module M2 {  
  language("C");  
  command("M2");  
  machine(name="hios.inf.pucrs.br",alias="hios");  
  port {  
    int A : out;  
    int B : out;  
    control comeca : out;  
    control res: in;  
    int result : in;  }}  

```

Arquivo de Coordenação^(2/2)

```
net N1 { A(VHDL),A(M2) }  
net N2 { B(VHDL),B(M2) }  
net N3 { comeca(VHDL),comeca(M2) }  
net N4 { res(M2),res(VHDL) }  
net N5 { result(VHDL),result(M2) }
```



Tópicos

1. Introdução
2. Ambiente de Co-simulação Desenvolvido
 - 2.1. Estrutura Geral
 - 2.2. Arquivo de Coordenação
 - 2.3. Software
 - 2.4. Hardware
3. Estudo de Caso
4. Conclusões

Exemplo de Software

```
#include "csi.h"
#include <stdio.h>

void main(int argc, char *argv[])
{
    int x=2,result;
    int i;

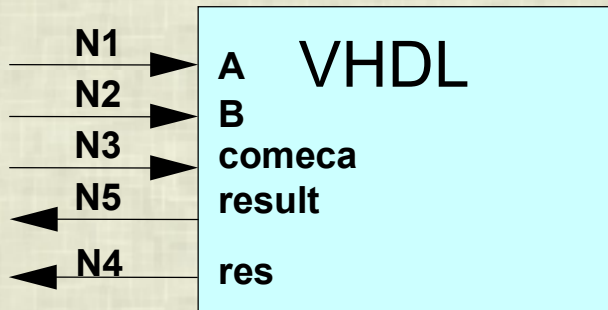
    csiInitialize();

    for(i=0;i<10;i++,x++) {
        csiSend("B",x);
        csiSend("A",x);
        csiSendControl("comeca");
        csiWaitControl("res");
        csiReceive("result",&result);
        printf("%d x %d = %d\n",x,x,result);
    }
    getchar();
}
```

Tópicos

1. Introdução
2. Ambiente de Co-simulação Desenvolvido
 - 2.1. Estrutura Geral
 - 2.2. Arquivo de Coordenação
 - 2.3. Software
 - 2.4. Hardware
3. Estudo de Caso
4. Conclusões

Exemplo de Hardware



```
.architecture tb_arch of vdhl_tb is
```

```
-- component CSI_IN
```

```
-- component CSI_OUT
```

```
-- component vhdl
```

```
.begin
```

```
-- Unit Under Test Port Map
```

```
UUT : vhdl port map();
```

```
-- Communication Components
```

```
ComLib1 : CSI_IN port map
```

```
(dout => a, porta => PORTA_A, clk => clk);
```

```
ComLib2 : CSI_IN port map (...);
```

```
ComLib3 : CSI_IN port map (...);
```

```
ComLib4 : CSI_OUT port map (...);
```

```
ComLib5 : CSI_OUT port map (...);
```

```
-- Clock and Reset Generation
```

```
end tb_arch;
```

Tópicos

1. Introdução

2. Ambiente de Co-simulação Desenvolvido

2.1. Estrutura Geral

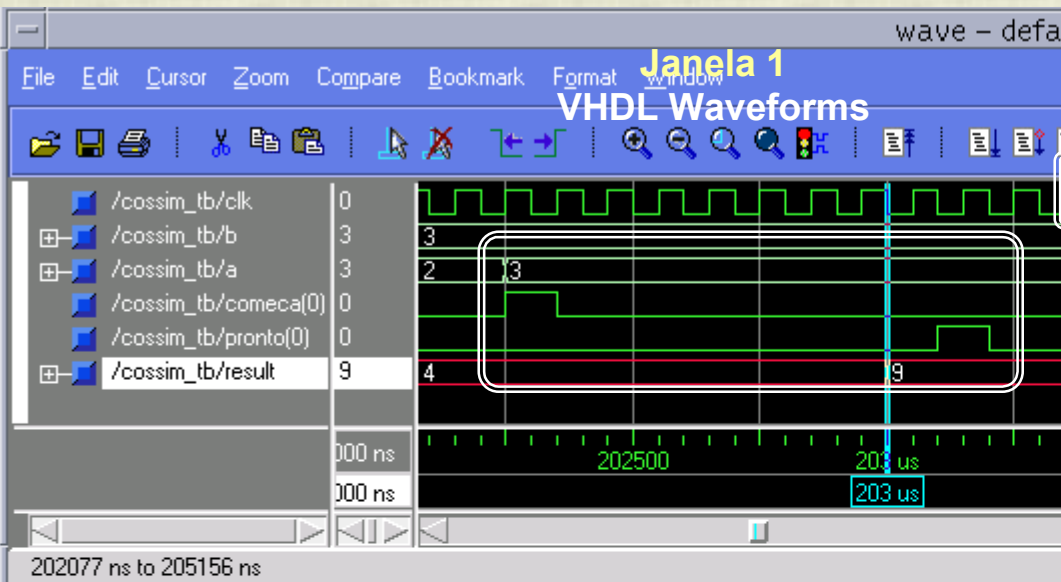
2.2. Arquivo de Coordenação

2.3. Software

2.4. Hardware

3. Estudo de Caso

4. Conclusões



Cosim-M2

```

Connecting to server hios.inf.pucrs.br at port 4233
Connection accepted
Sending control comeca
2 x 2 = 4
Sending control comeca
3 x 3 = 9
Sending control comeca
4 x 4 = 16
Sending control comeca
5 x 5 = 25
Sending control comeca
6 x 6 = 36
Sending control comeca
7 x 7 = 49
Sending control comeca
8 x 8 = 64
Sending control comeca
9 x 9 = 81
Sending control comeca
10 x 10 = 100
Sending control comeca
11 x 11 = 121
Shutting down CSI...

```

Janela 2 - M2

```

127 amory@hios:~/cst_new/cst/multi % ../bin/cst multi.conf
Server started at port 4233
Launching the clients...
VHDL
M2
Waiting for clients...
Client arriving...M2 ==> connected!
Client arriving...VHDL ==> connected!
*** All clients connected ***
Routing messages ...
exit
Exiting...
Total time to process messages is: 0min0s3ms
Messages routed: 100
Total time spent to simulate is: 5min19s449ms
N1 = 10 N2 = 10 N3 = 10 N4 = 10 N5 = 10
128 amory@hios:~/cst_new/cst/multi % █

```

Janela 3
Mensagens do
Backplane

ModelSim SE PLUS 5.5e

```

# Receiving control comeca
# Sending control res
# Receiving control comeca
# Sending control res
# Receiving control comeca
# Sending control res

```

VSIM 2>

Now: 1 ms Delta: 1 sim./cossim_tb

Janela 4
simulador de
hardware



Mensagens do Roteador

cst multi.conf

Server started at port 4233

Launching the clients...

VHDL

M2

Waiting for clients...

Client arriving...M2 ==> connected!

Client arriving...VHDL ==> connected!

*** All clients connected ***

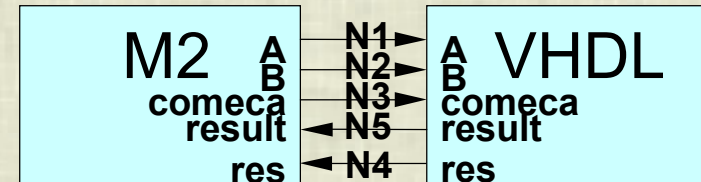
Routing messages ...

Total time to process messages is: 0min0s7ms

Messages routed: 100

Total time spent to simulate is: 23min18s572ms

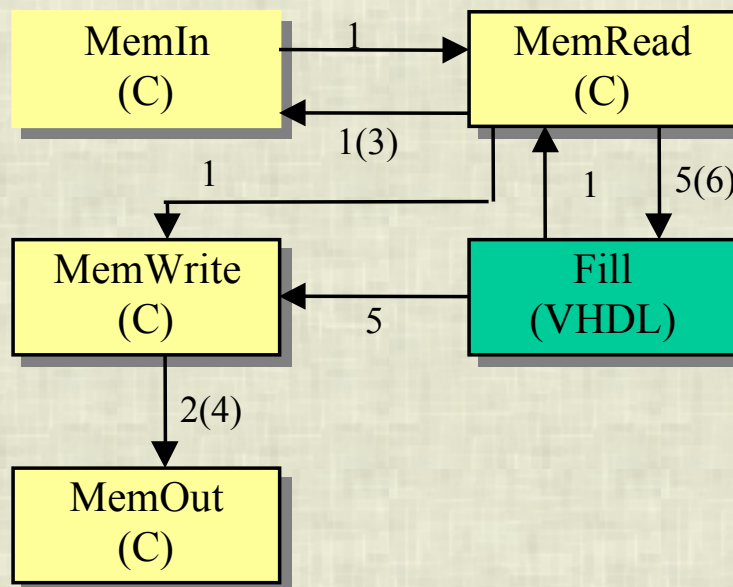
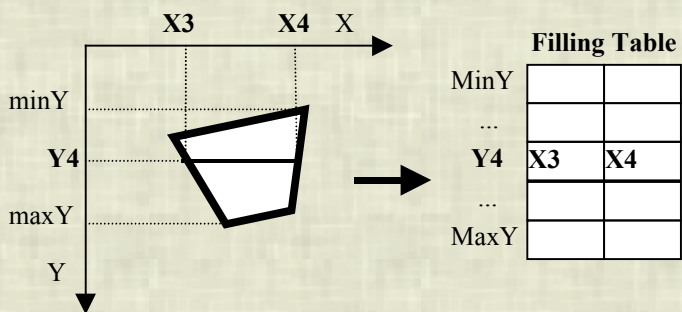
N1 = 10 N2 = 10 N3 = 10 N4 = 10 N5 = 10



```
net N1 { A(VHDL),A(M2) }
net N2 { B(VHDL),B(M2) }
net N3 { comeca(VHDL),comeca(M2) }
net N4 { res(M2),res(VHDL) }
net N5 { result(VHDL),result(M2) }
```

Preenchedor de Polígonos

- Composto por 5 módulos
- Partição possui grande tráfego de mensagens



Resultados Obtidos

# Polygons	CCC	SSVS	CO-SIM
50	0.002	4.4	34.8
100	0.003	4.9	69.4
500	0.017	9.4	352.8
1000	0.033	15.4	703.3
3000	0.100	40.0	2109.2

# Polygons	C + VHDL		C + VHDL (optim)	
	time	# msgs	time	# msgs
50	113.7	90495	34.8	56342
100	224.2	183527	69.4	114407
500	1120.9	924745	352.8	576312
1000	2160.3	1840089	703.3	1250308
3000	9608.2	5522387	2109.2	3755731

# Polygons	Single Machine	Distributed only LAN	Distributed LAN + WAN
50	34.8	41.0	36.3
100	69.4	76.4	72.5
500	352.8	423.1	368.1
1000	703.3	725.5	751.9
3000	2109.2	2177.4	2238.8

Tópicos

1. Introdução

2. Ambiente de Co-simulação Desenvolvido

2.1. Estrutura Geral

2.2. Arquivo de Coordenação

2.3. Software

2.4. Hardware

3. Estudo de Caso

4. Conclusões

Problemas da Ferramenta

- **Tempo de execução excessivo**
 - suportar métodos de comunicação mais rápidos (e.g. memória compartilhada) quando diversos módulos estiveram na mesma máquina
- **Não suporta simulação temporal**
- **Suporta somente tipo inteiro**
- **Estender a depuração do sistema**
- **Usuário deve informar o endereço do roteador nos módulos lançados remotamente**
 - suportar FTP para passar estes dados

Conclusões & Trabalhos Futuros

- **Conclusões**

- Suporta **co-simulação distribuída**
- Lança simuladores **automaticamente**
- Pode ser usada como ambiente de **validação de HDL** e **avaliação de particionamento**
- Suporta **VHDL** e **C**

- **Trabalhos Futuros**

- Suportar **SDL** e **Java**
- Geração automática de TB VHDL
- Otimizar tempo de execução
- Interface gráfica para criação do arquivo de coordenação