NOME:_____

No USP:_____

PSI-2553- Projeto de Sistemas Integrados

Folha de resultados da experiência:

O Processador Plasma - Parte 2

Compilação do projeto no ambiente Quartus da Altera e simulação do sistema Plasma (e seu processador LITE) usando o programa ModelSim e um *testbench*

Observação do arquivo plasma_tbw.vhd

- **Nomes dos sinais presentes no arquivo plasma_tbw.vhd:**

- **Valor do período de relógio:**

Resultados da compilação

- **código do componente: _____**

- **número de elementos lógicos utilizados: _____**

- **número de registradores utilizados: _____**

Resultados da simulação

Dica: modifique o tipo de apresentação dos sinais no ModelSim de binário para inteiro e hexadecimal (Radix) para facilitar a análise da simulação.

- **Identificar no ModelSim os nomes dos seguintes sinais:**

  1. **O registrador de instruções: _____**
  2. **O registrador do *opcode*: _____**
  3. **Os registradores do *Register file*: _____**
  4. **O endereço de escrita na memória: _____**
  5. **O valor dos dados: _____**

- **Quais registradores do *Register File* foram utilizados durante a simulação (anotar os números e nomes destes registradores)**

- **Quantas ocorrências de *write_enable* ocorreram ao longo de sua simulação? O que significam estas ocorrências?**

- **Anote os *opcodes*, endereços e instantes de tempo de simulação dos seguintes eventos:**

	T_simulação	**Opcode**	**Endereço**
Início da execução do Main			
Fim da execução do Main			
Início da execução da função Fibonacci			
Fim da execução da função Fibonacci			
Chamada da função Fibonacci no Main			
Passagem do argumento para a função Fibonacci			
Retorno do valor calculado na função Fibonacci para o Main			

Qual foi o tempo total de computação do número de Fibonacci?

Instante inicial: _____	**Opcode:_____**
Instante final: _____	**Opcode: _____**

APÊNDICE

1. Intel HEX-record Format (retirado de http://www.lucidtechnologies.info/)

INTRODUCTION

Intel's Hex-record format allows program or data files to be encoded in a printable (ASCII) format. This allows viewing of the object file with standard tools and easy file transfer from one computer to another, or between a host and target. An individual Hex-record is a single line in a file composed of many Hex-records.

HEX-RECORD CONTENT

Hex-Records are character strings made of several fields which specify the record type, record length, memory address, data, and checksum. Each byte of binary data is encoded as a 2-character hexadecimal number: the first ASCII character representing the high-order 4 bits, and the second the low-order 4 bits of the byte.

The 6 fields which comprise a Hex-record are defined as follows:

Field	Characters	Description
1	Start code	1
2	Byte count	2
3	Address	4
4	Type	2
5	Data	0-2n
6	Checksum	2

Each record may be terminated with a CR/LF/NULL. Accuracy of transmission is ensured by the byte count and checksum fields.

HEX-RECORD TYPES

There are three possible types of Hex-records.

00	A record containing data and the 2-byte address at which the data is to reside.
01	A termination record for a file of Hex-records. Only one termination record is allowed per file and it must be the last line of the file. There is no data field.
02	A segment base address record. This type of record is ignored by Lucid programmers.

HEX-RECORD EXAMPLE

Following is a typical Hex-record module consisting of four data records and a termination record.

```
:10010000214601360121470136007EFE09D2190140
:100110002146017EB7C20001FF5F16002148011988
:10012000194E79234623965778239EDA3F01B2CAA7
:100130003F0156702B5E712B722B732146013421C7
```

```
:00000001FF
```

The first data record is explained as follows:

: Start code.

10 Hex 10 (decimal 16), indicating 16 data character pairs, 16 bytes of binary data, in this record.

0100 Four-character 2-byte address field: hex address 0100, indicates location where the following data is to be loaded.

00 Record type indicating a data record.

The next 16 character pairs are the ASCII bytes of the actual program data.

40 Checksum of the first Hex-record.

The termination record is explained as follows:

: Start code.

00 Byte count is zero, no data in termination record.

0000 Four-character 2-byte address field, zeros.

01 Record type 01 is termination.

FF Checksum of termination record.