



Projeto

Definições

Agenda

→ Definições sobre o trabalho da disciplina





Trabalho da Disciplina

Sobre o multiplicador em ponto flutuante



multiplicador em ponto flutuante de 32 bits no padrão IEEE 754 (arquivo de topo “multiplier32FP”) com a saída registrada

Sobre o multiplicador em ponto flutuante



O circuito deve manipular números positivos e negativos e tratar/identificar os seguintes casos especiais:

- 1) positivo/negativo infinito (**infinif_o**): o expoente contém um padrão de bits reservado 11111111, a fração (mantissa) contém somente zeros, e o bit de sinal é 0 ou 1;
- 2) *not a number* (**nan_o**): o expoente contém um padrão de bits reservado 11111111, a fração (mantissa) é diferente de zero, e o bit de sinal é 0 ou 1. Neste caso, ambos operandos devem ser testados, a multiplicação não deve ser realizada e esta flag deve ir para '1'. Neste caso, o valor na saída deve ser 0x00000000;
- 3) multiplicar números classificados como “**zero sujo**”. Uma representação chamada de “zero sujo”, não-normalizada, permite representar números no intervalo entre 0 e o primeiro número representável na forma normalizada ($1,0 \times 2^{-126}$). O bit de sinal pode ser 0 ou 1 e o expoente contém o padrão de bits 00000000. A fração contém o padrão de bits real para a magnitude do número, em vez da mantissa. Deste modo, não existe nenhum 1 escondido neste formato. Números denormalizados, portanto, permitem que os números em ponto flutuante atinjam valores muito menores, sacrificando a quantidade de bits no significando;
- 4) **arredondamento**: *round toward zero* (arredonda em direção a zero): neste caso os bits que estão a mais são desprezados.
- 5) *overflow* (**overflow_o**): ocorre quando o expoente resultante excede o valor máximo permitido para este número normalizado. Neste caso, o valor na saída deve ser 0x7FFFFFFF;
- 6) *underflow* (**underflow_o**): devolve um número menor que o permitido normalizado. O *underflow* ocorre quando uma operação é executada e retorna um valor que é menor que o menor número não zero.
 - a. Sobre *underflow*: No padrão IEEE 754 precisão simples isto significa um valor que tem a magnitude (valor absoluto) menor que $1,0 \times 10^{-149}$ (número denormalizado). Normalmente quando um número chega a este patamar de magnitude ele é arredondado para zero, o que pode não fazer muita diferença em uma adição, mas tem um grande efeito na multiplicação. Neste caso, o valor na saída deve ser 0x00000000;

Sobre o multiplicador em ponto flutuante



Interface de topo do multiplicador

Nome	Tamanho	Tipo	Conexão	Descrição
clk	1	entrada	externa	Clock global do circuito
rst_n	1	entrada	externa	Reset global do circuito ativo em baixo
a_i	32	entrada	externa	Dado a ser multiplicado
b_i	32	entrada	externa	Dado a ser multiplicado
product_o	32	saída	externa	Resultado da multiplicação
start_i	1	entrada	externa	Indica quando deve ser iniciada uma nova operação de multiplicação
done_o	1	saída	externa	Indica que a multiplicação terminou e o valor na saída é válido
nan_o	1	saída	externa	<i>flag</i> para indicar que um operando que não é um número
infini_o	1	saída	externa	<i>flag</i> para indicar que um operando é infinito
overflow_o	1	saída	externa	<i>flag</i> para indicar que o resultado da operação gerou <i>overflow</i>
underflow_o	1	saída	externa	<i>flag</i> para indicar que o resultado da operação gerou <i>underflow</i>

Sobre o multiplicador em ponto flutuante



Observações adicionais

Circuito:

- **rst_n**: não utilizar para a operação regular de multiplicação do circuito;
- **done_o**: manter ativo (em alto) por 1 ciclo de clock;
- **nan_o**: manter ativo (em alto) por no máximo 2 ciclos de clock;
- **infinif_o**: manter ativo (em alto) por no máximo 2 ciclos de clock;
- **overflow_o**: manter ativo (em alto) por no máximo 2 ciclos de clock;
- **underflow_o**: manter ativo (em alto) por no máximo 2 ciclos de clock.

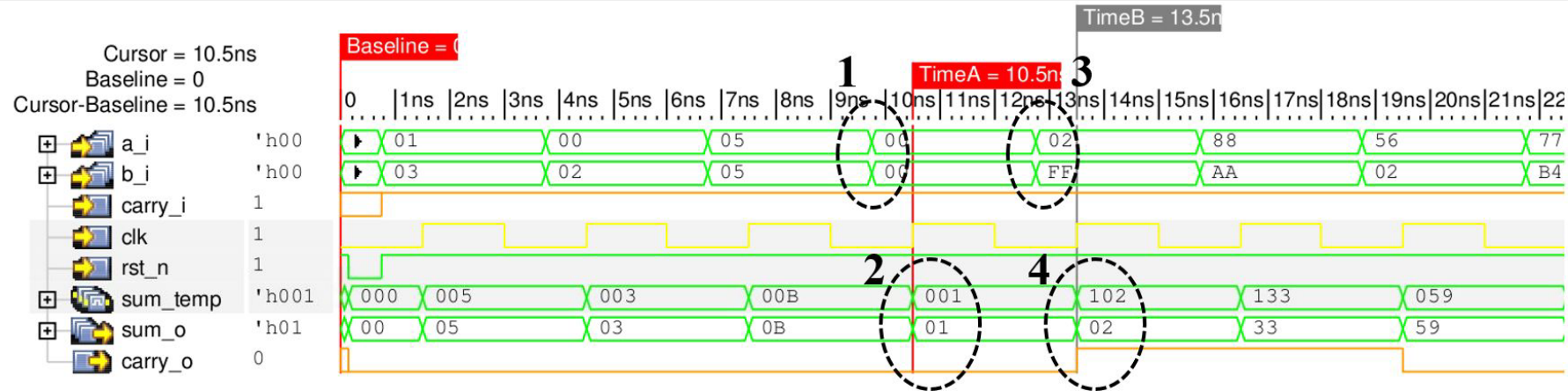
Testbench:

- o sinal para indicar o início de uma nova multiplicação, porta de entrada “**start_i**”, deve ser acionado no *testbench* dois (2) ciclos de clock depois da *flag* “**done_o**” indicar o fim da multiplicação. Para a multiplicação inicial, “start_i” deve ser verdadeiro **10 ciclos de clock após o início da simulação**.



Realizar as seguintes tarefas sobre o circuito

1) Simulação em nível HDL (mostrar o correto funcionamento do circuito, comentar as formas de onda conforme exemplo abaixo);



1 – entrada dos dados “a_i” e “b_i” no circuito somador um pouco antes da borda do clock em 10,5 ns.

2 – resultado do registrador temporário “sum_temp” em 9 bits, da saída “sum_o” em 8 bits e do “carry_o”. Repare que o resultado é 01h apenas em decorrência do “carry_i”.

3 – entrada dos dados “a_i” e “b_i” no circuito somador um pouco antes da borda do clock em 13,5 ns.

4 – o resultado do registrador temporário “sum_temp” em 9 bits mostra o MSB=1 o que se reflete na saída “carry_o”; o resultado “sum_o” mostra um valor que não é correto considerando a soma em representação direta.

Realizar as seguintes tarefas sobre o circuito



2) Síntese lógica (*constraints*);

Realizar as seguintes tarefas sobre o circuito



- 2) Síntese lógica (*constraints*);
- 3) Simulação *netlist* com SDF e geração de VCD pré-layout (mostrar o correto funcionamento do circuito, comentar as formas de onda conforme anterior);

Realizar as seguintes tarefas sobre o circuito



- 2) Síntese lógica (*constraints*);
- 3) Simulação *netlist* com SDF e geração de VCD pré-layout (mostrar o correto funcionamento do circuito, comentar as formas de onda conforme anterior);
- 4) Preenchimento de tabela e análise dos resultados (tabela a seguir);

Realizar as seguintes tarefas sobre o circuito



- 2) Síntese lógica (*constraints*);
- 3) Simulação *netlist* com SDF e geração de VCD pré-layout (mostrar o correto funcionamento do circuito, comentar as formas de onda conforme anterior);
- 4) Preenchimento de tabela e análise dos resultados (tabela a seguir);
- 5) Síntese física e obtenção de dados de desempenho;

Realizar as seguintes tarefas sobre o circuito



- 2) Síntese lógica (*constraints*);
- 3) Simulação *netlist* com SDF e geração de VCD pré-layout (mostrar o correto funcionamento do circuito, comentar as formas de onda conforme anterior);
- 4) Preenchimento de tabela e análise dos resultados (tabela a seguir);
- 5) Síntese física e obtenção de dados de desempenho;
- 6) Simulação *netlist* com SDF e geração de VCD pós-layout (mostrar o correto funcionamento do circuito, comentar as formas de onda conforme anterior). Para esta tarefa o profissional pode escolher um SDF pós-layout, desde que devidamente identificado/caracterizado;

Realizar as seguintes tarefas sobre o circuito



- 2) Síntese lógica (*constraints*);
- 3) Simulação *netlist* com SDF e geração de VCD pré-layout (mostrar o correto funcionamento do circuito, comentar as formas de onda conforme anterior);
- 4) Preenchimento de tabela e análise dos resultados (tabela a seguir);
- 5) Síntese física e obtenção de dados de desempenho;
- 6) Simulação *netlist* com SDF e geração de VCD pós-layout (mostrar o correto funcionamento do circuito, comentar as formas de onda conforme anterior). Para esta tarefa o profissional pode escolher um SDF pós-layout, desde que devidamente identificado/caracterizado;
- 7) Obtenção de dados de desempenho: área, *timing*, *power*, conforme tabela.

Lista de arquivos digitais a serem entregues



Compactar toda a pasta do projeto (**multiplier32FP**) em um único arquivo (.zip), o qual contém toda a árvore de diretórios utilizada, bem como os arquivos internos solicitados. Dentro da pasta de topo deve haver um “readme.txt” instruindo como executar o *script* de síntese. O arquivo compactado deve ser entregue via *upload* no moodle. Não incluir arquivos temporários gerados pelas ferramentas, apenas os arquivos solicitados.

Lista de arquivos digitais a serem entregues



Compactar toda a pasta do projeto (**multiplier32FP**) em um único arquivo (.zip), o qual contém toda a árvore de diretórios utilizada, bem como os arquivos internos solicitados. Dentro da pasta de topo deve haver um “readme.txt” instruindo como executar o *script* de síntese. O arquivo compactado deve ser entregue via *upload* no moodle. Não incluir arquivos temporários gerados pelas ferramentas, apenas os arquivos solicitados.

O relatório em formato pdf deve ser carregado em separado.

Lista de arquivos digitais a serem entregues



Compactar toda a pasta do projeto (**multiplier32FP**) em um único arquivo (.zip), o qual contém toda a árvore de diretórios utilizada, bem como os arquivos internos solicitados. Dentro da pasta de topo deve haver um “readme.txt” instruindo como executar o *script* de síntese. O arquivo compactado deve ser entregue via *upload* no moodle. Não incluir arquivos temporários gerados pelas ferramentas, apenas os arquivos solicitados.

O relatório em formato pdf deve ser carregado em separado.

(checklist) Lista final de arquivos para fazer *upload*:

<nome>.zip

relatorio.pdf

Constraints



```
set MAIN_CLOCK_NAME clk
```

```
set MAIN_RST_NAME rst_n
```

```
set BEST_LIB_OPERATING_CONDITION PVT_1P32V_0C
```

```
set WORST_LIB_OPERATING_CONDITION PVT_0P9V_125C
```

```
set period_clk ????? ;#clk = ????? MHz = ??? ns (period)
```

```
set clk_uncertainty 0.044 ;# ns ("a guess")
```

```
set clk_latency 0.105 ;# ns ("a guess")
```

```
set in_delay 0.28 ;# ns
```

```
set out_delay 0.35 ;#ns
```

```
set out_load 0.045 ;#pF
```

```
set slew "146 164 264 252" ;#minimum rise, minimum fall, maximum rise and maximum fall
```

```
set slew_min_rise 0.146 ;# ns
```

```
set slew_min_fall 0.164 ;# ns
```

```
set slew_max_rise 0.264 ;# ns
```

```
set slew_max_fall 0.252 ;# ns
```

Constraints



```
set MAIN_CLOCK_NAME clk
set MAIN_RST_NAME rst_n
set BEST_LIB_OPERATING_CONDITION PVT_1P32V_0C
set WORST_LIB_OPERATING_CONDITION PVT_0P9V_125C
```

set period_clk ????? ;#clk = ????? MHz = ??? ns (period)

```
set clk_uncertainty 0.044 ;# ns ("a guess")
```

```
set clk_latency 0.105 ;# ns ("a guess")
```

```
set in_delay 0.28 ;# ns
```

```
set out_delay 0.35 ;# ns
```

```
set out_load 0.045 ;# pF
```

```
set slew "146 164 264 252" ;# minimum rise, minimum fall, maximum rise and maximum fall
```

```
set slew_min_rise 0.146 ;# ns
```

```
set slew_min_fall 0.164 ;# ns
```

```
set slew_max_rise 0.264 ;# ns
```

```
set slew_max_fall 0.252 ;# ns
```

**2 frequências: 10 MHz e
a maior “possível”**

Constraints



```
set MAIN_CLOCK_NAME clk
set MAIN_RST_NAME rst_n
set BEST_LIB_OPERATING_CONDITION PVT_1P32V_0C
set WORST_LIB_OPERATING_CONDITION PVT_0P9V_125C
```

set period_clk ?????? ;#clk = ?????? MHz = ??? ns (period)

```
set clk_uncertainty 0.044 ;# ns ("a guess")
```

```
set clk_latency 0.105 ;# ns ("a guess")
```

```
set in_delay 0.28 ;# ns
```

```
set out_delay 0.35 ;# ns
```

```
set out_load 0.045 ;# pF
```

```
set slew "146 164 264 252" ;# minimum rise, minimum fall, maximum rise and maximum fall
```

```
set slew_min_rise 0.146 ;# ns
```

```
set slew_min_fall 0.164 ;# ns
```

```
set slew_max_rise 0.264 ;# ns
```

```
set slew_max_fall 0.252 ;# ns
```

**2 frequências: 10 MHz e
a maior “possível”**

```
e32000d6 632000d6
c72360d6 c72360d6
472360d6 c72360d6
7edfe8ea 0ad828eb
de3d104f 9e2117ff
bcc001f4 fccf01f4
3f800000 c75ff506
9f2e99d6 4ed088ea
1d8647c1 004de8c6
ff8364ef 93ea69c6
ce9981fd 2e5948b7
f58705e4 9affd518
8847902a 50d33667
93c34f89 2aa4a1c6
4553b6c9 3d7694bf
a813c7c2 9bd93e71
7b9f4594 b81df364
e0a7d0b4 df6e5b3e
10f05801 483be748
29224825 e9f94cee
ca14a237 1ce51fc8
226fb519 7fc8e1cb
abd762f3 f0d653d6
8fd2a024 382ab6e3
a04fbe18 2f6e5923
eeeaba0e 16c3b505
```

Usar o arquivo “vetor.txt” (disponibilizado pelo Professor) com 100 vetores de teste para todas as etapas do fluxo



Arquivos a serem enviados

multiplier32FP	<- diretório recebe o nome da entidade de topo do projeto ({DESIGNS})
-- frontend	<- HDL's sintetizáveis, <i>testbench</i> , sdf_cmd_file.cmd, sdf_cmd_file_layout.cmd, arquivos VCD, vetor.txt, xrun.log, xmelab.log, xmsdfc.log
-- backend	
-- synthesis	
-- constraints	<- {DESIGNS}.sdc
-- deliverables	<- {DESIGNS}.v e {DESIGNS}_normal_worst.sdf (um arquivo para cada frequência)
-- reports	<- {DESIGNS}_{area,qor,gates,timing}.rpt (um arquivo para cada frequência)
-- scripts	<- run_first.tcl, {DESIGNS}.tcl
-- common	<- path.tcl, sdf_width_wa.etf, tech.tcl
-- work	<- genus.log, genus.cmd (últimos arquivos criados!)
-- layout	
-- constraints	<- sem arquivos
-- deliverables	<- {DESIGNS}_layout.def, {DESIGNS}_layout.sdf, {DESIGNS}_layout.v (um arquivo para cada frequência)
-- reports	<- ccop-log.rpt, power_no_VCD.rpt, power_VCD_MAX.rpt, power_VCD_MIN.rpt (um arquivo para cada frequência)
-- scripts	<- cts.ccopt, layout.tcl, {DESIGNS}.view
-- work	<- innovus.cmd, innovus.log, todo o conteúdo do diretório "to-drc-lvs.enc" (um arquivo para cada frequência)
-- verification	<- (apenas circuito com maior frequência)
-- work	<- cds.lib, assura_tech.lib, diretórios na íntegra ("assura", "runDRC", "runLVS", {DESIGNS}), defin.log, verilogn.log, libManager.log

Caso o arquivo VCD seja muito grande, informar o tamanho em MBytes do mesmo no relatório e, apenas neste caso, não é necessário entregá-lo via *upload* no moodle.

Garantir que todos os **arquivos de log solicitados** sejam os últimos gerados (sugestão: verificar a opção de sobrescrever quando da invocação de cada ferramenta).

Arquivos a serem enviados



Observações adicionais

SDF:

- Utilize a seguinte nomenclatura para a geração do SDF:
- Síntese **Lógica**: **multiplier32FP_normal_worst.sdf**
 - o compilado: **multiplier32FP_normal_worst.sdf.X**
- Síntese **Física**:
 - multiplier32FP_normal_layout_MAX.sdf**
 - o compilado: **multiplier32FP_normal_layout_MAX.sdf.X**
 - multiplier32FP_normal_layout_MIN.sdf**
 - o compilado: **multiplier32FP_normal_layout_MIN.sdf.X**

Tabela de desempenho





Tabela de desempenho

Freq (MHz)	# of vectors	VCD simulation time (ns)	Power (μW)			Total area (μm^2)	# of gates	Timing slack (ps)	Start-point End-point	TNS (ps)
			internal	switching	total					
10	--	--								
	100	X								
			2X							
?	--	--								
	100	X								
			2X							

Tabela de desempenho



OBS 1: Usar 4 casas após a vírgula para expor os resultados

OBS 2: O número de *gates* equivalentes é igual a à área total do circuito dividida pela área de uma porta NAND2_{PL} de 2 entradas da tecnologia em questão e *performance level* igual a “A”.

OBS 3: Para a informação de *timing slack* da tabela, referenciar no documento o *print* do relatório entregue pela ferramenta, tanto para a síntese lógica quanto para a síntese física. Deve estar visível o caminho crítico.

OBS 4: Dispor todas as informações em uma única tabela, como apresentado.

Recomendações gerais



- * Concordâncias verbal e nominal foram corrigidas?
- * Erros de grafia/digitação foram corrigidos?
- * Uso excessivo das mesmas palavras em uma frase ou parágrafo foram evitados?
- * Frases longas demais, com muitas vírgulas e encadeando diversos assuntos e/ou conclusões foram evitadas?
- * Todas as bibliografias que estão citando livros foram escritas mencionando a página ou as páginas pesquisadas para obter a informação?
- * Palavras em outra língua estão em *itálico*? Palavras de uso comum como chip, software, clock ou hardware, apesar de serem estrangeiras, não necessitam ser grafadas em *itálico*, justamente pelo grau de inserção na língua Portuguesa.
- * Sempre que uma sigla for apresentada pela primeira vez no texto, mesmo que o trabalho conte com uma lista de abreviaturas, a mesma deve ter seu significado expresso por extenso.
- * Índices e siglas para variáveis em fórmulas devem seguir uma uniformidade (letra em maiúsculo/minúsculo) em todo o texto. Esta uniformidade deve considerar texto, equações, figuras, esquemáticos, fluxos, ilustrações...
- * Verifique: se estiver escrevendo um texto em Português, a separação decimal dos números é feita com a vírgula e não com o ponto.
- * Para gráficos: 1) não repetir (o título já está apresentado na legenda da figura) o título do gráfico na imagem que foi capturada do Excel/Matlab/Mathcad/... ou qualquer outro software que foi usado para gerá-lo; 2) tenha certeza de que foram inseridas as identificações para os eixos x e y.
- * Para equações: confira se após apresentá-la, os seus símbolos estão descritos na linha seguinte, sem parágrafo.
- * Todas as equações, figuras, tabelas, quadros... devem estar numerados e devem ser chamados no texto.
- * Referências: apenas insira uma referência bibliográfica se ela possui citação em alguma parte do texto.

02/12/2024





- Estudar a teoria do processo de multiplicação em ponto flutuante 32 bits
- Projetar em SystemVerilog ou Verilog um multiplicador em ponto flutuante 32 bits (envolve o trabalho da disciplina a ser entregue em ~~07/02/2025~~ **19/02/2025** até às 23h)