



UNIVERSIDADE DE BRASÍLIA
Organização e arquitetura de computadores

Relatório Laboratório 2

Grupo 3

Davi Mansur Costa - 211042701
Hagatta Giovanna Nogueira de Sousa - 190088508
Leonardo Ramos Barbosa - 211010360
Maria Eduarda Alves de Sousa - 190113219
Pedro Brum Tristão de Castro - 202067470

2) (5.5) Unidade Lógico Aritmética de Inteiros:

Abra o Quartus Prime e Project/Restore Archived Project e escolha o arquivo ULA.qar

a) (1.0) Para a ULA de inteiros fornecida, analise sua descrição Verilog e descreva suas funções e escreva a tabela de seus códigos para cada operação. Defina o arquivo ALU.v como toplevel, compile e visualize o circuito sintetizado com o Tools/Netlist Viewers/RTL viewer.

A ULA de inteiros tem ao todo 20 funções com dois números (A e B) de 32 bits:

AND: realiza a operação AND bit a bit em dois números de 32 bits (A AND B)

OR: realiza a operação OR bit a bit em dois números de 32 bits (A OR B)

XOR: realiza a operação XOR bit a bit em dois números de 32 bits (A XOR B)

ADD: realiza a adição entre dois números de 32 bits (A + B)

SUB: realiza a subtração entre dois números de 32 bits (A - B)

SLT: verifica se $A < B$, ambos com sinal

SLTU: verifica se $A < B$, ambos sem sinal

SLL: desloca os bits do A, n bits para a esquerda, com n sendo os 5 bits menos significativos de B

SRL: desloca os bits do A, n bits para a direita, com n sendo os 5 bits menos significativos de B

SRA: desloca os bits do A, n bits para a direita, mantendo o sinal de A, com n sendo os 5 bits menos significativos de B

LUI: carrega B na saída

MUL: carrega os 32 bits menos significativos da multiplicação entre dois números de 32 bits (A * B)

MULH: carrega os 32 bits mais significativos da multiplicação entre dois números de 32 bits, ambos com sinal (A * B)

MULHU: carrega os 32 bits mais significativos da multiplicação entre dois números de 32 bits, ambos sem sinal (A * B)

MULHSU: carrega os 32 bits mais significativos da multiplicação entre dois números de 32 bits, considerando A com sinal e B sem sinal (A * B)

DIV: realiza a divisão entre dois números de 32 bits, ambos com sinal (A / B)

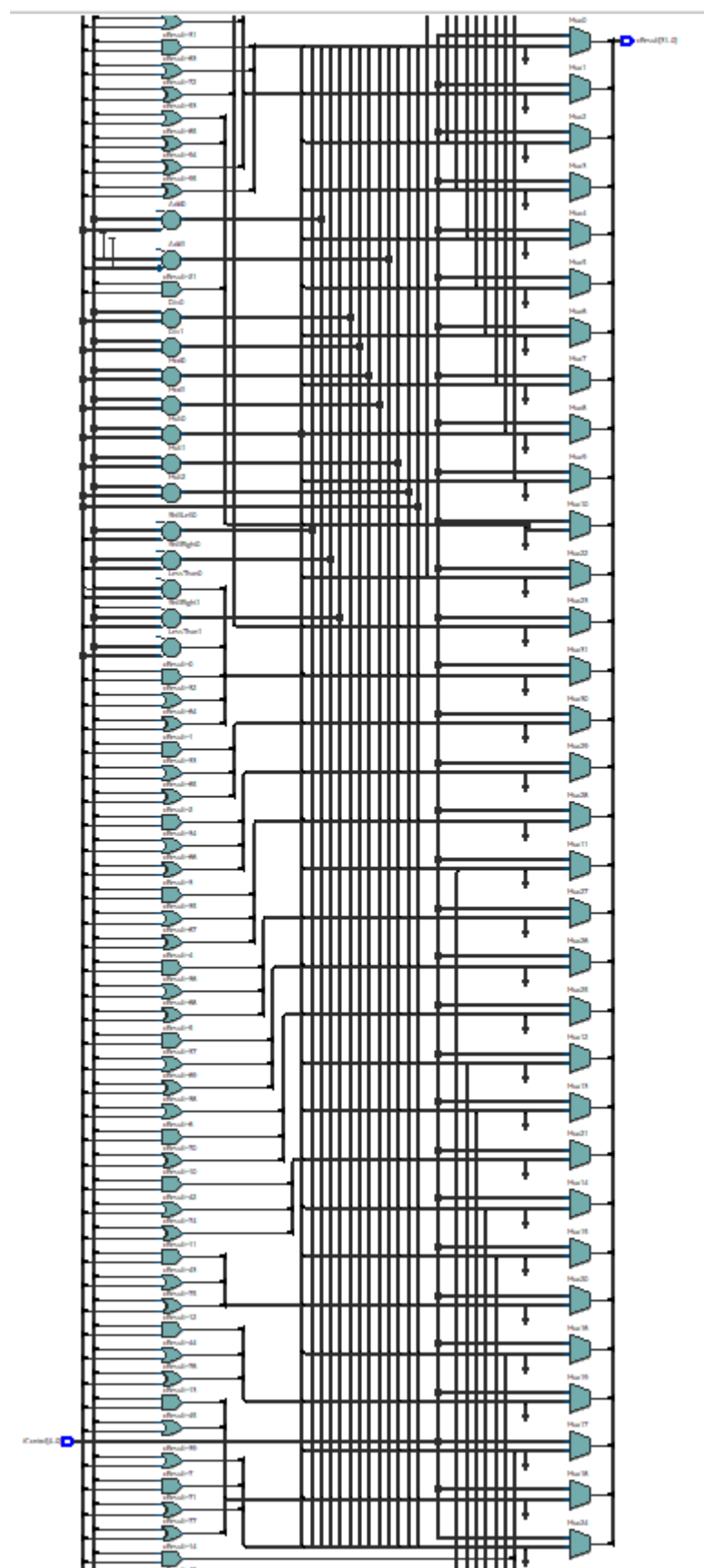
DIVU: realiza a divisão entre dois números de 32 bits, ambos sem sinal (A / B)

REM: realiza o módulo entre dois números de 32 bits, ambos com sinal (A % B)

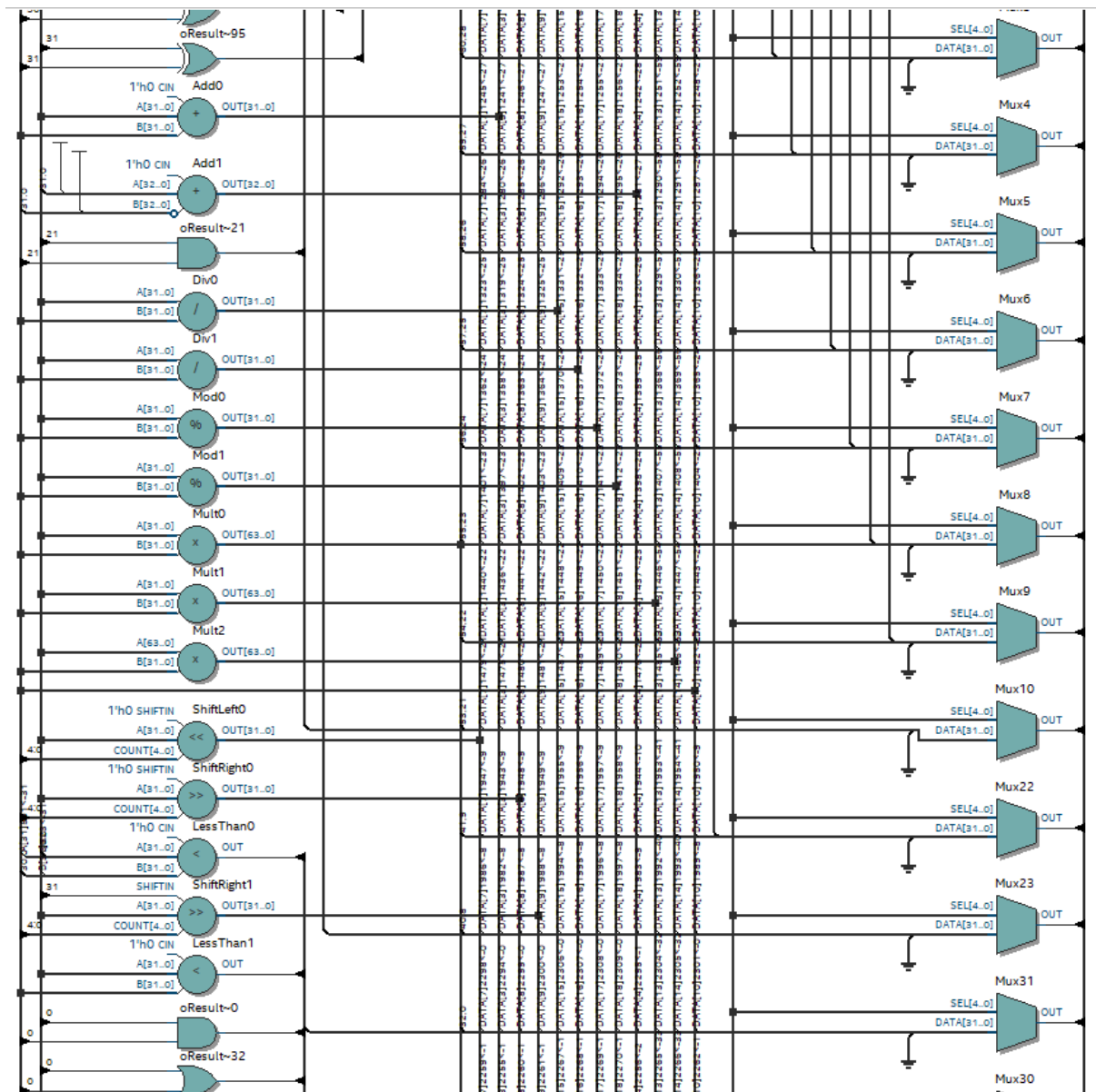
REMU: realiza o módulo entre dois números de 32 bits sem sinal, ambos sem sinal (A % B)

NULL: coloca 0 na saída

OPERAÇÃO	CÓDIGO
AND	00000
OR	00001
XOR	00010
ADD	00011
SUB	00100
SLT	00101
SLTU	00110
SLL	00111
SRL	01000
SRA	01001
LUI	01010
MUL	01011
MULH	01100
MULHU	01101
MULHSU	01110
DIV	01111
DIVU	10000
REM	10001
REMU	10010
NULL	11111



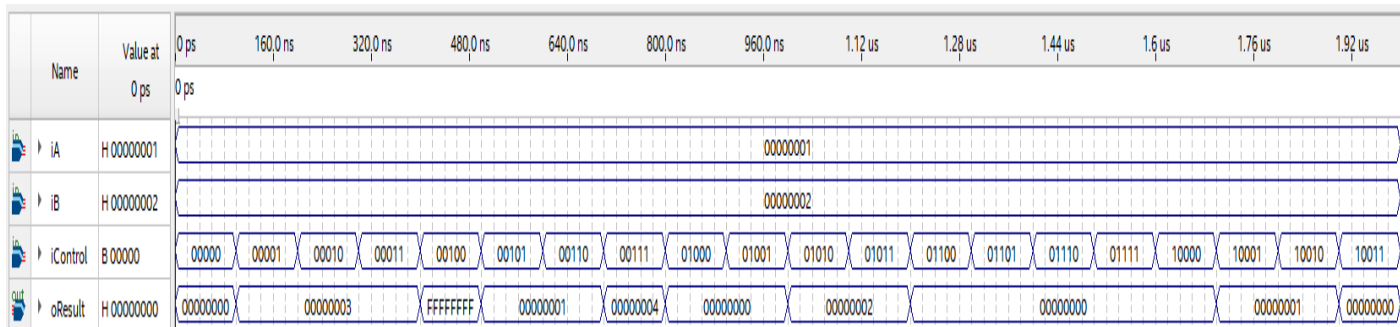
Circuito completo.



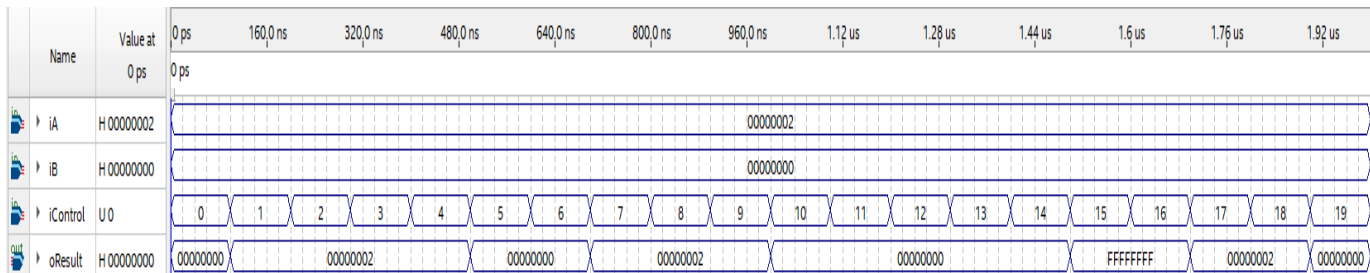
Mais detalhes da parte responsável pelas operações e alguns multiplexadores.

b) (1.0) Modifique os valores do arquivo de forma de onda ULA.vwf e verifique cada operação implementada. Dica: Escolha valores de entrada que sejam representativos (comuns) e valores que possam gerar resultados singulares (overflow, divisão por zero etc.). Tire print screens para colocar no relatório.

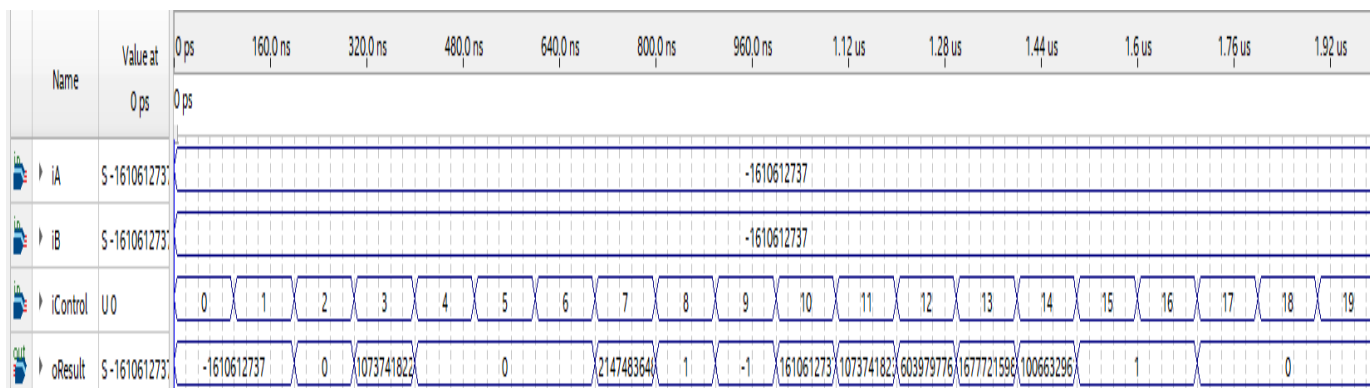
Primeiro escolhemos os valores comuns 1 e 2:



Testamos a divisão por 0 utilizando os valores 2 e 0:



Em seguida utilizamos valores para gerar overflow



c) (1.0) Indique os requisitos físicos da implementação da ULA total e para cada operação separadamente: i) Número de Elementos Lógicos (ALMs), ii) Número de Registradores e iii) Quantidade de bits de memória e iv) Número de blocos DSP usados. v) Somando todos os ALMs do item i) compare com o número de ALMs da ULA total e explique a diferença. vi) Indique quais são e analise o impacto das funções com maiores circuitos no tamanho da ULA. Dica: Defina manualmente o sinal de controle iControl no arquivo ULA.v, pois o Quartus otimiza o projeto retirando as partes não utilizadas.

OPERAÇÃO	ALMs	REGISTRADORES	BITS DE MEMÓRIA	DSP
----------	------	---------------	-----------------	-----

ULA TOTAL	3033	0	0	12
AND	65	0	0	0
OR	65	0	0	0
XOR	65	0	0	0
ADD	65	0	0	0
SUB	65	0	0	0
SLT	79	0	0	0
SLTU	79	0	0	0
SLL	120	0	0	0
SRL	121	0	0	0
SRA	118	0	0	0
LUI	49	0	0	0
MUL	56	0	0	2
MULH	72	0	0	3
MULHU	72	0	0	3
MULHSU	88	0	0	6
DIV	704	0	0	0
DIVU	633	0	0	0
REM	728	0	0	0
REMU	652	0	0	0
NULL	49	0	0	0

Somando o número das ALMs de cada operação, encontramos o valor 3945, que é superior ao número de ALMs utilizadas pela ULA completa, isso acontece pelo fato que o software Quartus otimiza o circuito.

As operações mais custosas são a DIV e REM e seus correspondentes sem sinal. Algoritmos de divisões são mais custosos por precisar calcular restos e quocientes, fazendo assim ser necessária uma grande quantidade de ALMs

d) (1.0) Usando o TimeQuest, indique os requerimentos temporais para a ULA total e para cada operação separadamente: i) o caminho de maior atraso, ii) maior tempo

de atraso tpd. iii) Indique quais são e analise o impacto das funções mais demoradas no tpd.

OPERAÇÃO	CAMINHO DE MAIOR ATRASO	MAIOR ATRASO TPD (ns)
ULA TOTAL	iB[7] para oResult[19]	105.323
AND	iA[31] para oResult[31]	2.312
OR	iA[31] para oResult[31]	2.312
XOR	iA[31] para oResult[31]	2.312
ADD	iB[2] para oResult[26]	2.865
SUB	iB[3] para oResult[24]	2.939
SLT	iA[18] para oResult[0]	4.490
SLTU	iA[18] para oResult[0]	4.490
SLL	iA[21] para oResult[25]	4.166
SRL	iB[1] para oResult[2]	4.237
SRA	iA[31] para oResult[5]	4.099
LUI	iB[29] para oResult[29]	0.516
MUL	iB[9] para oResult[28]	7.547
MULH	iA[18] para oResult[22]	8.165
MULHU	iA[16] para oResult[25]	8.535
MULHSU	iB[20] para oResult[27]	11.571
DIV	iB[0] para oResult[31]	92.628
DIVU	iB[26] para oResult[0]	81.512
REM	iB[7] para oResult[24]	94.185
REMU	iB[27] para oResult[12]	87.490
NULL	Nothing to report	

Como analisado no item acima, as operações REM e DIV são as mais custosas em número de ALMs e tem também o maior tpd pelo fato de serem mais complexas acabam tendo um tempo de atraso maior que as demais.

e) (1.0) Defina o arquivo TopDE.v como o toplevel, sintetize na DE1-SoC e filme comprovando o correto funcionamento de acordo com os valores testados no item b).

Link para o vídeo: <https://youtu.be/4aVyBDv4dl0>

f) (0.5) Verifique o quanto seu grupo consegue otimizar esta ULA em termos de requerimentos físicos e/ou temporais (sem retirar nenhuma funcionalidade!).

3) (5.5) Unidade Aritmética de Ponto Flutuante:

Abra o Quartus Prime e Project/Restore Archived Project e escolha o arquivo FPALU.qar

a) (1.0) Para a FPULA fornecida, analise sua descrição Verilog e descreva suas funções e escreva a tabela de seus códigos para cada operação. Defina o arquivo FPALU.v como toplevel, compile e visualize o circuito sintetizado com o Tools/Netlist Viewers/RTL viewer.

A ULA de ponto flutuante tem 20 funções com dois números de 32 bits (A e B):

ADD: realiza a adição entre dois números de 32 bits em ponto flutuante ($A + B$)

SUB: realiza a subtração entre dois números de 32 bits em ponto flutuante ($A - B$)

MUL: realiza a multiplicação entre dois números de 32 bits em ponto flutuante ($A * B$)

DIV: realiza a divisão entre dois números de 32 bits em ponto flutuante (A / B)

SQRT: calcula a raiz quadrado de A

ABS: faz o valor absoluto de A

CEQ: verifica se $A = B$

CLT: verifica se $A < B$

CLE: verifica se $A \leq B$

CVTSW: converte um número de inteiro com sinal para ponto flutuante

CVTWS: converte um número em ponto flutuante para inteiro com sinal

MV: coloca o número A na saída

SGNJ: coloca o sinal do número B em A

SGNJN: coloca o inverso do sinal do número B em A

SGNJX: verifica se A e B têm sinais diferentes, se sim A fica negativo, senão A fica positivo

MAX: coloca o maior número na saída

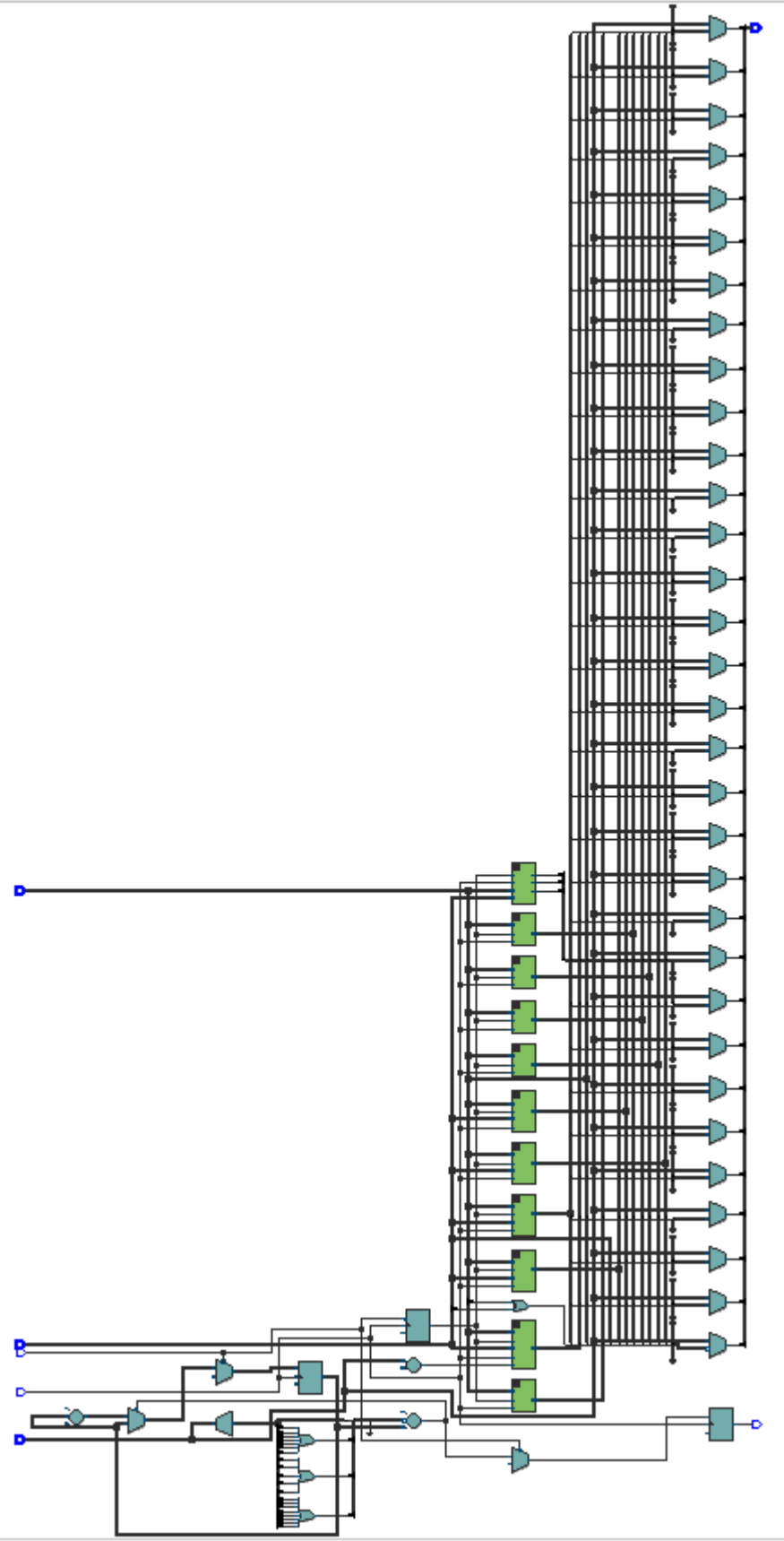
MIN: coloca o menor número na saída

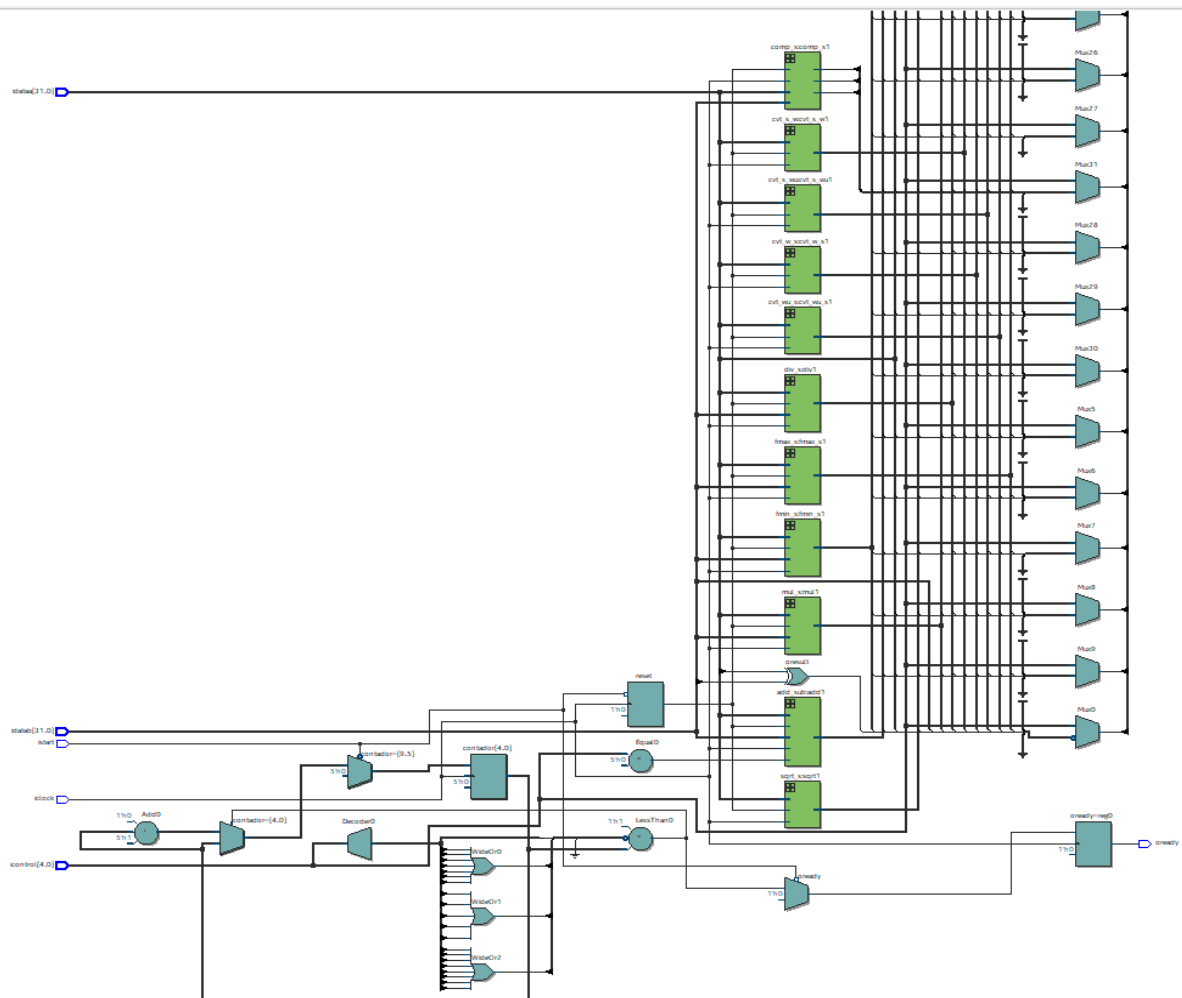
CVTSWU: converte um número inteiro sem sinal para ponto flutuante

CVTWUS: converte um número em ponto flutuante para inteiro sem sinal

NULL: coloca 0xEEEEEEEE na saída

OPERAÇÃO	CÓDIGO
ADD	00000
SUB	00001
MUL	00010
DIV	00011
SQRT	00100
ABS	00101
CEQ	00110
CLT	00111
CLE	01000
CVTSW	01001
CVTWS	01010
MV	01011
SGNJ	01100
SGNJD	01101
SGNDX	01110
MAX	01111
MIN	10000
CVTSWU	10001
CVTWUS	10010
NULL	11111

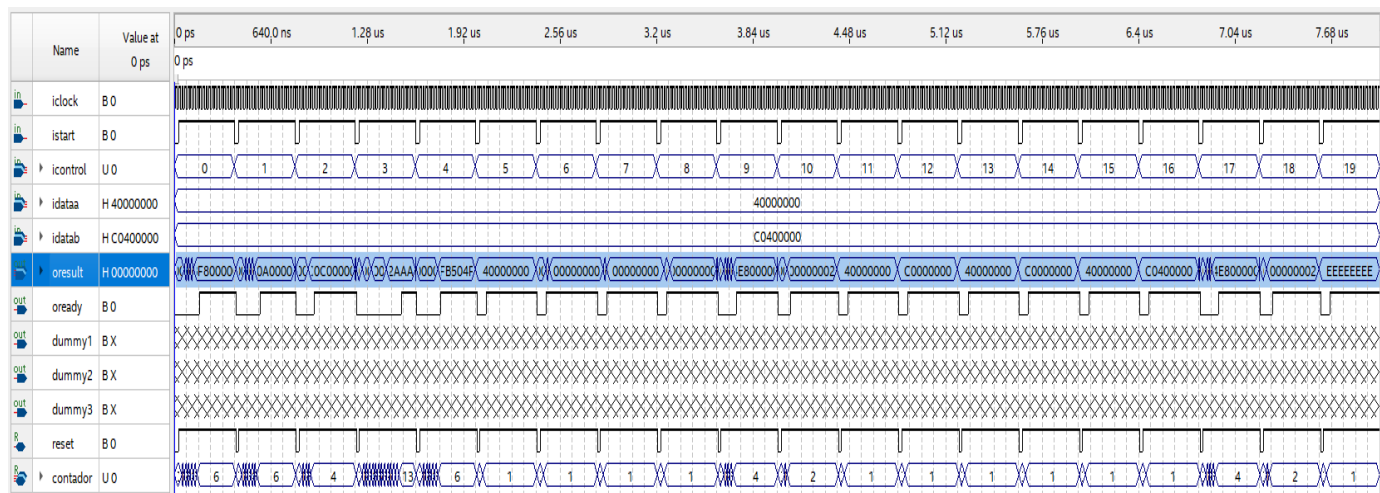




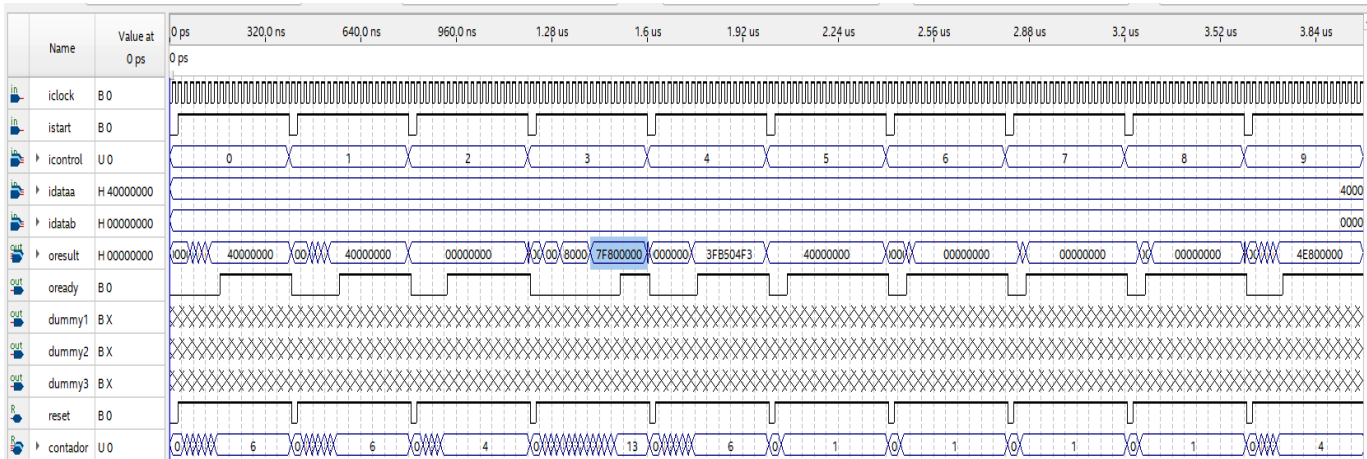
b) (1.0) Modifique os valores do arquivo de forma de onda FPULA.vwf e verifique cada operação implementada.

Dica: Escolha valores de entrada que sejam representativos (comuns) e também valores que gerem resultados singulares (divisão por zero, overflow, underflow, NaN etc.). Tire print screens para colocar no relatório.

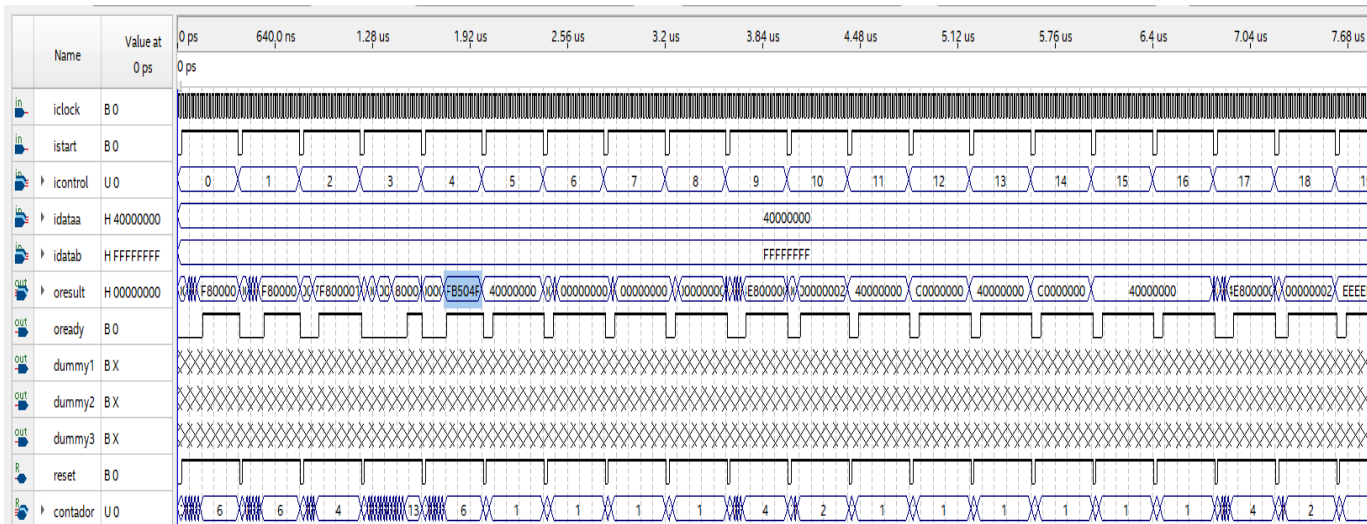
Teste com valores comuns:



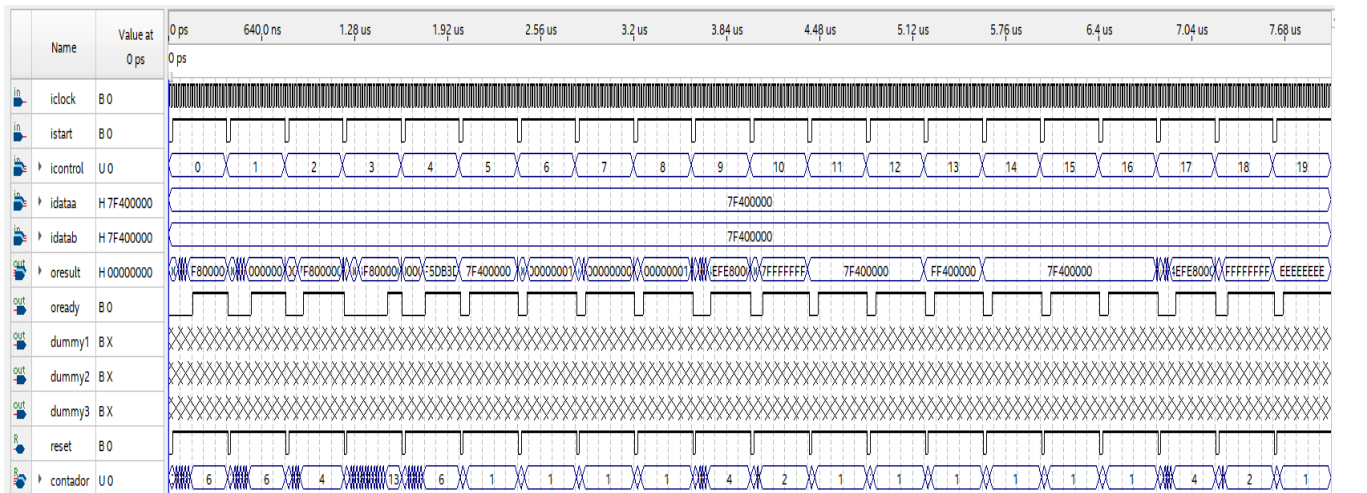
Divisão por 0:



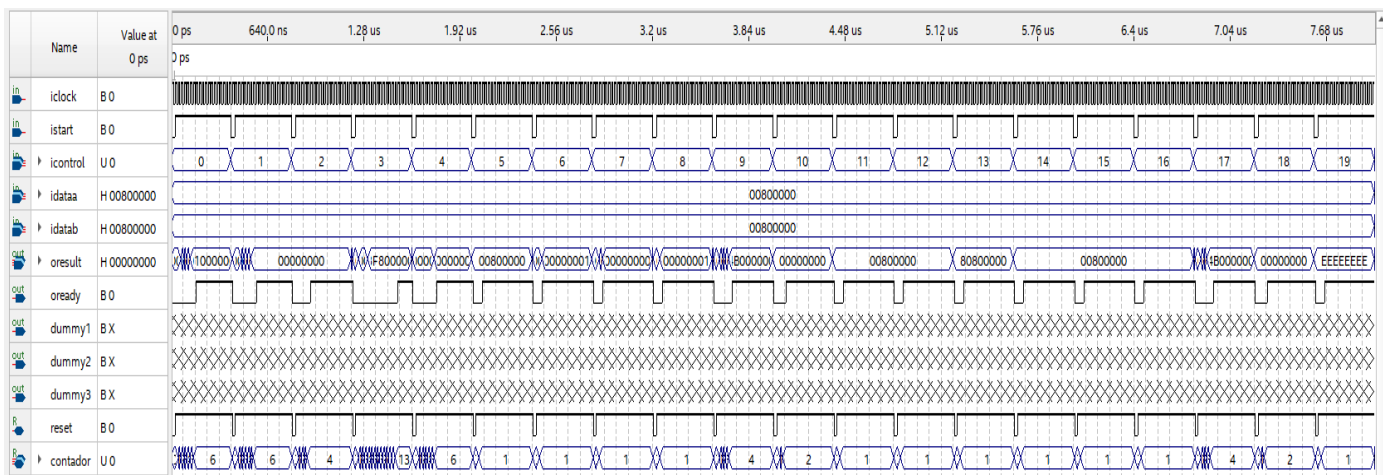
NaN:



Overflow:



Underflow:



c) (1.0) Indique os requisitos físicos da implementação da FPULA total e para cada operação separadamente: i) Número de Elementos Lógicos (ALMs), ii) Número de Registradores e iii) Quantidade de bits de memória e iv) Número de blocos DSP usados, v) Somando todos os ALMs do item i) compare com o número de ALMs da ULA total e explique a diferença. vi) Indique quais são e analise o impacto das funções com maiores circuitos no tamanho da ULA.
Dica: Defina manualmente o icontrol pois o Quartus otimiza o projeto retirando os elementos não utilizados.

OPERAÇÃO	ALMs	REGISTRADORES	BITS DE MEMÓRIA	DSP
FPULA TOTAL	1321	1096	47616	6
ADD	388	288	0	0
SUB	382	287	0	0
MUL	116	77	0	1
DIV	206	288	31744	3
SQRT	126	129	15872	2
ABS	49	2	0	0
CEQ	79	24	0	0
CLT	89	30	0	0
CLE	87	30	0	0
CVTSW	188	150	0	0
CVTWS	202	80	0	0

MV	49	2	0	0
SGNJ	49	2	0	0
SGNJN	49	2	0	0
SGNJX	50	2	0	0
MAX	97	2	0	0
MIN	97	2	0	0
CVTSWU	160	111	0	0
CVTWUS	188	43	0	0
NULL	49	2	0	0

Somando o número de ALMs utilizadas por cada operação separadamente encontramos 2700 ALMs, bem maior que a quantidade total da FPULA. Isso se dá pela otimização feita pelo Quartus. Na FPULA as instruções com maior impacto são o ADD e SUB, essa complexidade se deve à necessidade de igualar os expoentes dos operandos e tratar os bits com excedentes da mantissa, fazendo assim que os circuitos sejam mais elaborados e complexos que os feitos com números inteiros.

d) (1.0) Usando o TimeQuest, defina um clock de 50MHz, indique os requerimentos temporais para a ULA total e para cada operação separadamente: i) número de ciclos necessários à execução de cada operação, ii) tempos t_h , t_{co} , t_{su} , iii) máxima frequência de clock utilizável, e iv) se há algum requerimento não atendido. Indique quais são e analise o impacto das funções mais demoradas na máxima frequência utilizável. Aumente a frequência para 200MHz e repita os itens i), ii), iii) e iv). Analise os resultados obtidos.

50MHz

OPERAÇÃO	Nº DE CICLOS	TH	TCO	TSU	FREQ. MÁX
FPULA TOTAL		3.812	16.832	6.040	152.51 MHz
ADD	4	3.584	8.290	4.535	189.18 MHz
SUB	4	3.566	8.022	4.852	189.93 MHz
MUL	3	4.096	8.490	0.992	169.89 MHz
DIV	9	3.722	9.119	2.787	153.75 MHz
SQRT	7	3.644	10.924	0.585	251.7 MHz
ABS	1	1.756	7.311	-0.718	1148.11 MHz

CEQ	1	3.487	7.943	0.506	809.06 MHz
CLT	1	3.355	9.422	3.227	685.4 MHz
CLE	1	3.515	7.677	1.052	670.24 MHz
CVTSW	3	3.491	8.669	2.816	230.52 MHz
CVTWS	2	3.511	9.322	2.099	186.08 MHz
MV	1	1.627	7.300	-0.456	1149.43 MHz
SGNJ	1	1.627	7.300	-0.456	1149.43 MHz
SGNJN	1	1.627	7.300	-0.456	1149.43 MHz
SGNJX	1	1.305	9.645	0.082	1154.73 MHz
MAX	1	1.521	9.357	-0.117	1146.79 MHz
MIN	1	1.521	9.357	-0.117	1146.79 MHz
CVTSWU	3	3.554	11.538	1.139	238.66 MHz
CVTWUS	2	3.147	9.757	4.704	859.85 MHz
NULL	1	1.615	7.310	-0.446	1148.11 MHz

200MHz

OPERAÇÃO	Nº DE CICLOS	TH	TCO	TSU	FREQ. MÁX
FPULA TOTAL		3.812	16.832	6.040	152.51 MHz
ADD	16	3.584	8.290	4.535	189.18 MHz
SUB	16	3.566	8.022	4.852	189.93 MHz
MUL	7	4.096	8.490	0.992	169.89 MHz
DIV	15	3.722	9.119	2.787	153.75 MHz
SQRT	10	3.644	10.924	0.585	251.7 MHz

ABS	1	1.756	7.311	-0.718	1148.11 MHz
CEQ	2	3.487	7.943	0.506	809.06 MHz
CLT	4	3.355	9.422	3.227	685.4 MHz
CLE	4	3.515	7.677	1.052	670.24 MHz
CVTSW	12	3.491	8.669	2.816	230.52 MHz
CVTWS	6	3.511	9.322	2.099	186.08 MHz
MV	1	1.627	7.300	-0.456	1149.43 MHz
SGNJ	1	1.627	7.300	-0.456	1149.43 MHz
SGNJJN	1	1.627	7.300	-0.456	1149.43 MHz
SGNJX	1	1.305	9.645	0.082	1154.73 MHz
MAX	3	1.521	9.357	-0.117	1146.79 MHz
MIN	3	1.521	9.357	-0.117	1146.79 MHz
CVTSWU	10	3.554	11.538	1.139	238.66 MHz
CVTWUS	5	3.147	9.757	4.704	859.85 MHz
NULL	1	1.615	7.310	-0.446	1148.11 MHz

A partir das tabelas, observa-se que a operação que necessita da menor frequência é a divisão, a qual limita a frequência máxima da FPALU, além dela outras operações matemáticas (ADD, SUB, MUL) assim como CVTSWU também exigem frequências menores que 200 MHz para funcionar, logo essas funções não operam corretamente ao colocar a frequência de 200 MHz. Outro ponto importante de se observar é que o número de ciclos para cada operação aumenta com o aumento da frequência.

e) (1.0) Defina o arquivo TopDE.v como o toplevel, sintetize na DE1-SoC e filme comprovando o correto funcionamento de acordo com os valores de entrada testados no item b).

Link para o vídeo: <https://youtu.be/4aVyBDv4dl0>

f) (0.5) Verifique o quanto seu grupo consegue otimizar esta FPULA em termos de requerimentos físicos e/ou temporais (sem retirar nenhuma funcionalidade!).