

UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ
CIÊNCIA DA COMPUTAÇÃO

JOÃO PEDRO ROJAS, RAFAEL MENDONÇA VAZ, VINÍCIUS ALMEIDA HERMES

RELATÓRIO TRABALHO 1ª AVALIAÇÃO

CASCADEL

2025

1. Introdução

Este relatório documenta o processo de construção de um circuito operacional capaz de realizar as operações de soma e subtração de duas palavras em 8 bits, como forma de avaliação parcial da primeira nota para a disciplina de Sistemas Digitais, ministrada pela professora MSc. Gabriela Stein.

2. Requisitos

O circuito do somador completo deve operar a soma e a subtração de duas palavras de 8 bits, tendo a escolha da operação definida por um sinal 'op', que quando definido como '0', deve somar, e quando definido como '1', a operação a ser realizada é a subtração. Outro requisito para o circuito é a utilização do 'bit-vector' para os sinais e portas de dados da interface, o que permite a manipulação direta dos bits, isso significa que toda a lógica deve ser implementada bit a bit. Além disso, deve contar também com sinais de Carry-in, Carry-out e Overflow que são usados para indicar, respectivamente, o vai-um inicial, o vai-um final da operação e se houve estouro aritmético no resultado.

3. Descrição das situações

A implementação consiste em um circuito que realiza a operação de soma para A e B, independentemente de qual operador ('op') for escolhido, afinal a subtração é a soma de um número negativo ($A + \sim B$). Nisso, surge o primeiro desafio: quando estamos trabalhando com bits, obtemos apenas valores de 0 e 1, que retornam apenas números positivos. Portanto, para representação de um número assinalado, é necessário o uso do complemento de dois, que consiste em utilizar o algarismo de maior valor da combinação binária para "dizer" aritmeticamente se o valor representado é positivo ou negativo.

Uma vez tendo a opção de obter o número negativo em binário para realizar a soma no caso de subtração, é preciso escolhê-lo para "enviar" o valor correto, e para isso será necessária a utilização de um multiplexador de dois canais por 8 bits (MUX 2x8), que consiste em um circuito capaz de retornar um sinal dentre duas entradas

através de um seletor, que definirá se o B escolhido será o B ou o $\sim B$ para realizar o complemento de dois. Esse circuito equivale ao funcionamento de oito MUX 2x1 em paralelo, cada um escolhendo o respectivo bit com base no sinal de seleção.

Uma vez selecionado o valor de $\sim B$ quando o sinal 'op' for 1, a operação de soma é realizada por meio de oito somadores de 1 bit conectados em paralelo. O sinal de carry-in inicial (Cin_geral) é utilizado para completar o complemento de dois, sendo igual a 1 na subtração ('op' = 0) e 0 na adição ('op' = 1). Dessa forma, o circuito utiliza a estrutura do somador, aproveitando o carry-in para implementar corretamente a subtração sem a necessidade de circuitos adicionais.

Com base nesta análise, o sistema requer a utilização de 4 componentes principais para a sua implementação:

- Somador completo de 1 bit, realiza a operação básica binária;
- Somador completo de 8 bits, conecta 8 somadores de 1 bit em cascata;
- Inversor de 8 bits, responsável por inverter os bits de B para realizar o complemento de 2.
- Multiplexador 2x8, seleciona entre B e $\sim B$, baseado no sinal da operação.

3.1 Somador completo de 1 bit

Realiza a soma de dois bits (A e B) e o Carry-In resultando em uma soma (S) e um Carry-out.

Tabela verdade para o somador completo de 1 bit:

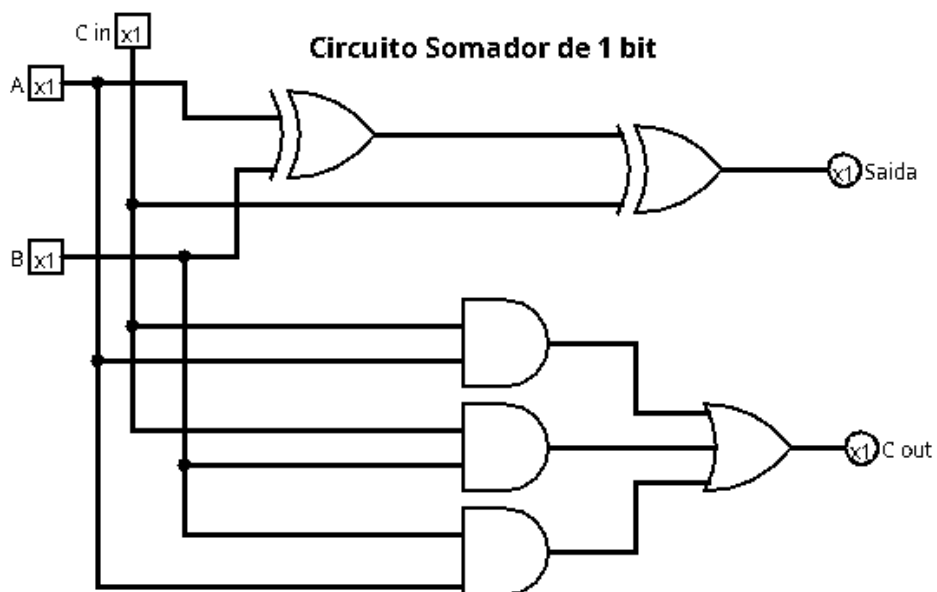
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Expressão booleana extraída da tabela:

$$\text{Soma (S)} = (A \oplus B) \oplus \text{Cin}$$

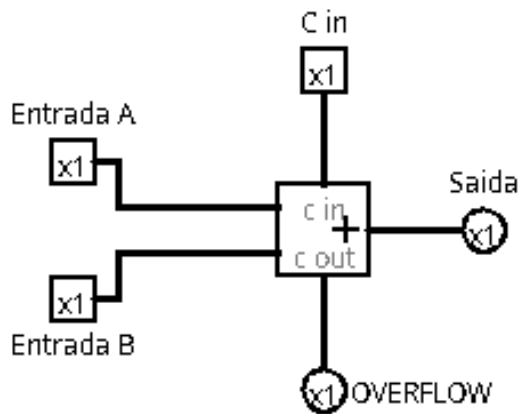
$$\text{Carry-out (Cout)} = (A \cdot B) + (B \cdot \text{Cin}) + (A \cdot \text{Cin})$$

Circuito para o somador completo de 1 bit:



Módulo somador completo de 1 bit:

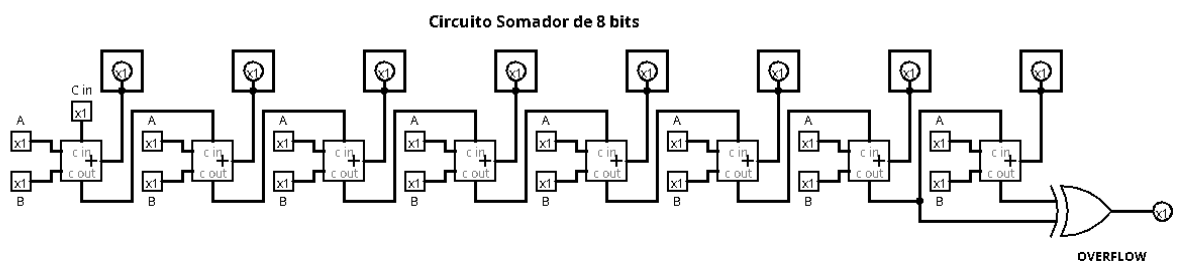
Somador de 1 bits



3.2 Somador completo de 8 bits

Realiza a soma de palavras de 8 bits através de 8 somadores completos de 1 bit ligados em paralelo, nos quais o sinal Carry-out é enviado ao Carry-in do próximo, o Carry-out do somador do bit de maior valor representa o Carry-out geral.

Circuito somador completo de 8 bits:



3.3 Inversor de 8 bits (bitwise)

Realiza a operação booleana NOT para cada bit da palavra de 8 bits. Como o inversor opera de forma independente bit-a-bit, é necessário 1 inversor de 1 bit para cada bit da palavra, neste caso, 8.

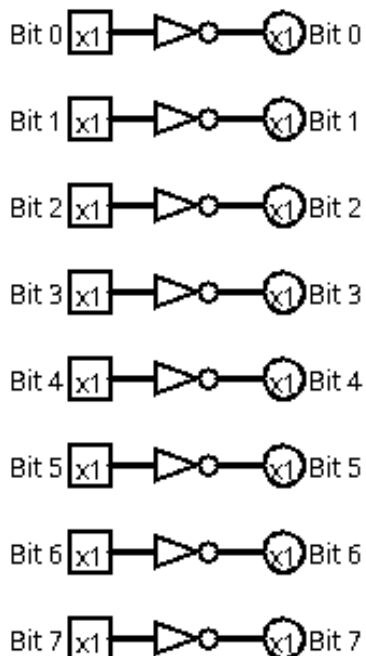
Tabela verdade do inversor de 1 bit:

B	Saída
0	1
1	0

Expressão booleana: Saída = $\sim B$

Circuito do inversor de 8 bits:

Circuito Inversor de 8 bits



3.4 Multiplexador 2x8

3.4.1 Multiplexador 2x1

Para a construção de um MUX2x8 são necessários 8 MUX2x1 ligados em paralelos. Criação do MUX de 2 canais por 1 bit:

Descrições:

Canais: Canal 0: {0}; Canal 1: {1};

Sinais do seletor: $\log_2 2 = 1$, portanto o seletor será de 1 bit;

Saída: Z de 1 bit. $Z = \{\text{Canal 0}, \text{Canal 1}\}$;

Comportamento: Select {0} comuta Canal 0 para saída Z;

Select {1} comuta Canal 1 para saída Z;

Tabela-verdade para o MUX2x1:

Canal 0	Canal 1	Seletor	Saída Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

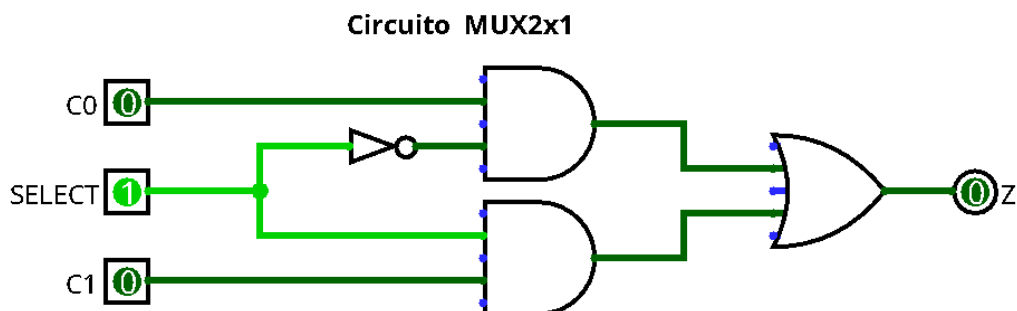
Expressão booleana:

$$Z = (\sim C0 \cdot C1 \cdot \text{SELECT}) + (C0 \cdot \sim C1 \cdot \sim \text{SELECT}) + (C0 \cdot C1 \cdot \sim \text{SELECT}) + (C0 \cdot C1 \cdot \text{SELECT})$$

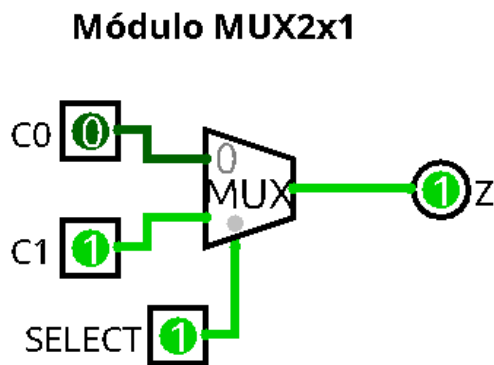
Simplificação:

$$Z = (\sim S \cdot C0) + (S \cdot C1)$$

Circuito para o MUX2x1:



Módulo MUX2x1:



3.4.2 Criação do MUX2x8

Descrições:

Canais: Canal 0: {C00, C01*, C02, C03, C04, C05, C06, C07};

Canal 1: {C10, C11, C12, C13, C14, C15, C16, C17};

*Notação: C01 = Canal 0, bit 1

Sinais do seletor: $\log(2)$ na base 2 = 1, portanto o seletor será de 1 bit;

Saída: Z de 1 bit. Z = {Canal 0, Canal 1};

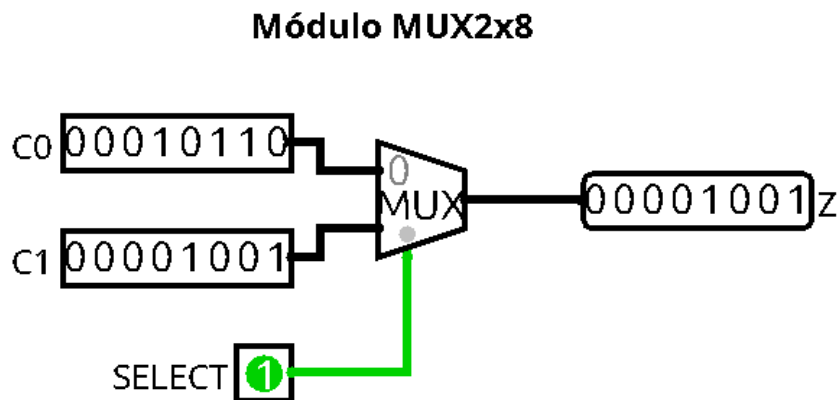
Comportamento: Select {0} comuta Canal 0 para saída Z;

Select {1} comuta Canal 1 para saída Z;

Tabela-verdade para o MUX2x8 (simplificada):

Canal 0	Canal 1	Seletor	Saída Z
C0[bit7-0]	C1[bit7-0]	0	C0[bit7-0]
C0[bit7-0]	C1[bit7-0]	1	C1[bit7-0]

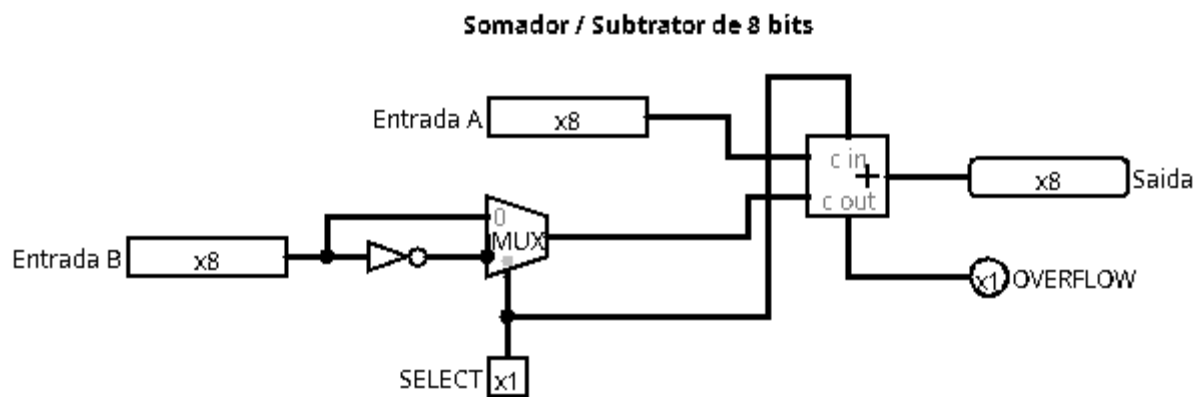
Módulo para o MUX2x8:



3.5 Somador/subtrator de 8 bits

O somador/subtrator de 8 bits opera com um MUX2x8 para a seleção de operação com base no sinal 'op' que é enviado como seletor para o MUX.

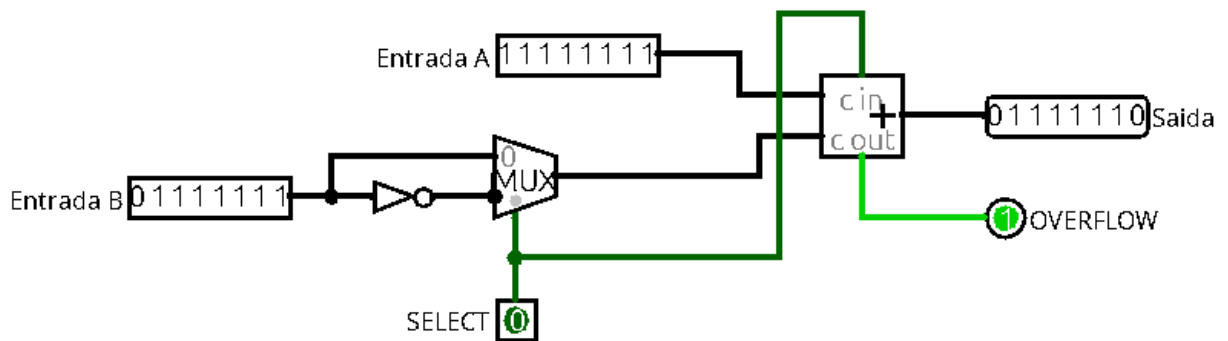
Circuito do somador/subtrator de 8 bits:



3.5.1 Somador

Caso o 'op' for '0' (soma), o MUX seleciona o sinal B envia-o para o somador, assim, será feita a soma de $A + B$.

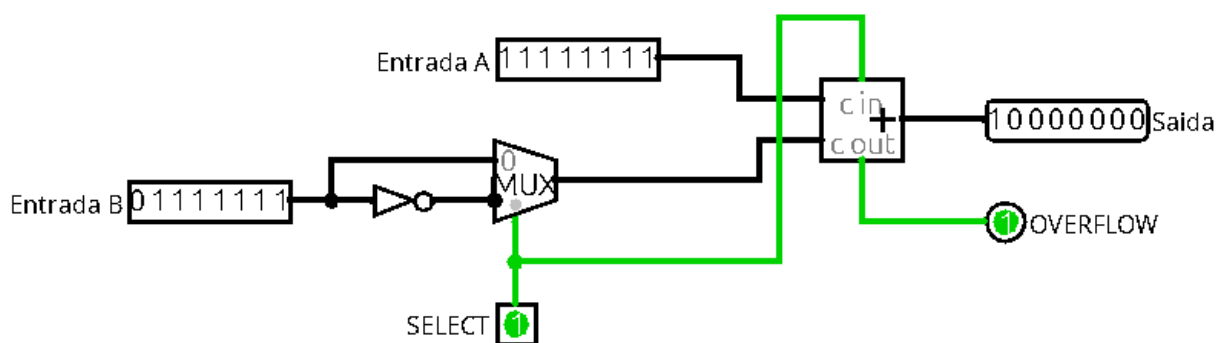
Funcionamento do circuito de soma:



3.5.2 Subtrator

Caso o 'op' for '1' (subtração), o MUX seleciona o sinal $\sim B$, e envia-o para o somador, o sinal do seletor ('1') também é enviado como carry-in no circuito de soma, aproveitando o sinal para fazer o complemento de dois no $\sim B$. Assim, será feita a soma de $A + (\sim B + 1)$.

Funcionamento do circuito de subtração:



4. Simulações

O circuito foi implementado em VHDL e foram realizadas duas simulações de soma ou subtração de valores hexadecimais representados em binário.

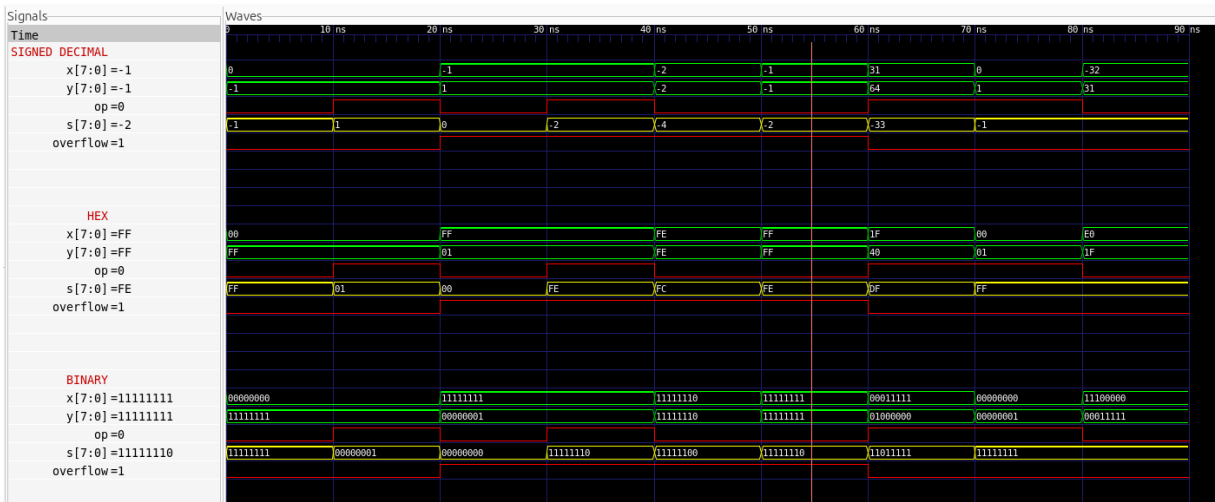
Tabela de casos testados:

A (HEX)	A (BIN)	OPERAÇÃO	B (HEX)	B (BIN)	RES (HEX)	RES (BIN)	OVERFLOW
0x00	0000 0000	+	0xFF	1111 1111	0xFF	1111 1111	0
0x00	0000 0000	-	0xFF	1111 1111	0x01	0000 0001	0
0xFF	1111 1111	+	0x01	0000 0001	0x00	0000 0000	1
0xFF	1111 1111	-	0x01	0000 0001	0xFE	1111 1110	1
0xFE	1111 1110	+	0xFE	1111 1110	0xFC	1111 1100	1
0xFF	1111 1111	+	0xFF	1111 1111	0xFE	1111 1110	1
0x1F	0001 1111	-	0x40	0100 0000	0xDF	1101 1111	0
0x00	0000 0000	-	0x01	0000 0001	0xFF	1111 1111	0
0xE0	1110 0000	+	0x1F	0001 1111	0xFF	1111 1111	0

4.1 Simulação 1

A primeira simulação foi realizada de modo que o circuito apenas realize a operação selecionada, para melhor visualização foi escolhido o tempo de 10 ns para cada sinal. Os valores foram formatados para uma melhor visualização dos resultados, sendo representados em decimal com sinal, hexadecimal e binário.

Representação visual da simulação 1 no software GTKWAVE:



4.2 Simulação 2

A segunda simulação foi realizada com uma latência de 4ns por porta lógica, de modo que o circuito tenha um atraso para realizar cada operação, desta forma,

foi percebido que o sinal fica com o valor correto apenas no final do tempo estipulado para cada sinal. Os valores também foram formatados em decimal com sinal, hexadecimal e binário.

4.2.1 Sinal

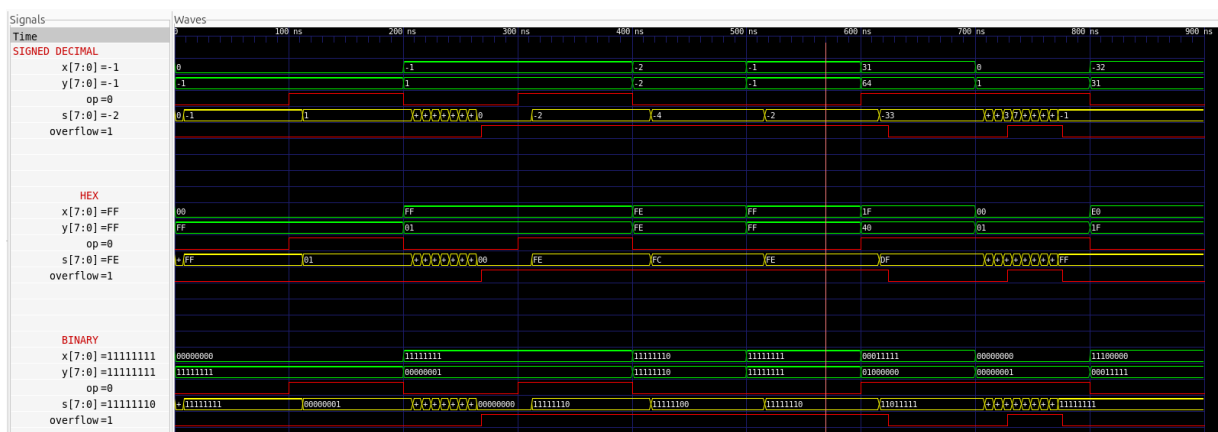
O valor escolhido para cada sinal foi de 100ns, dado por: atraso total = atraso combinacional + (N × atraso do somador)

Onde:

- Atraso combinacional = Inversor + MUX = 4 + 4 = 8ns
- N = 8 bits
- Atraso do somador de 1 bit = 8ns (2*4ns)
- Atraso total = 8 + (8 × 8) = 8 + 64 = 72 ns

Somamos ao atraso total uma margem de segurança de 40% para que ao final do atraso, de tempo de que todos os valores sejam representados corretamente.

Representação visual da simulação 2 no software GTKWAVE:



5. Conclusão

O processo de programação do trabalho apresentou pequenas dificuldades iniciais devido a necessidade de um aprendizado e melhor entendimento da sintaxe de arquivos .vhdl para a parte de programação. Isso foi observado durante a construção das simulações um e dois.

Na simulação um, foi possível reafirmar diversos conceitos que foram passados nas aulas teóricas e práticas. Através do uso de multiplexador para a construção de somadores e subtratores, com palavras de 8 bits, tendo o seletor como definidor do tipo de operação a ser realizada, ajudou a firmar os conhecimentos passados em aula.

Na simulação dois, percebemos que o tempo de sinal precisou ser aumentado para que a operação pudesse ser completada e pudesse ser observada, pois o atraso de 4 ns implicou em um atraso crescente das outras operações, totalizando cerca de 72 ns. Optamos por adicionar uma margem de segurança de 40% para garantir que o resultado correto aparecesse em tela.

Concluimos que a realização deste trabalho foi de suma importância para melhor fixação do conhecimento de circuitos digitais, portas lógicas, multiplexadores e somadores/subtratores. Reiteramos que a aplicação de trabalhos similares, que servem como complemento a avaliações são também, excelentes ferramentas de ensino.