ES572 - Circuitos Lógicos

Resumo Teórico

5 de setembro de 2021

Guilherme Nunes Trofino
217276

Conteúdo

1. Introdução

Apresentação Neste documento será descrito as informações necessárias para compreensão e solução de exercícios relacionados a disciplina 1.0.0.0. Note que este documento são notas realizadas por Guilherme Nunes Trofino, em 5 de setembro de 2021.

1.1. Informação

Definição Informação são dados comunicados ou recebidos que resolvem incertezas sobre um fato ou circunstância específica. Assim, dada uma variável aleatória discreta x com as seguintes condições:

1. Possíveis Valores: $x \in \{x_1, ..., x_n\}$;
2. Probabilidades Associadas: $\{p_1, ..., p_n\}$;

Desta forma, considera-se $I(x_i)$ que a **Quantidade de Informação Recebida**, medida em `bits`, será relacionada por:

$$\boxed{I(x_i) = \log_2\left(\frac{1}{p_i}\right)} \tag{1.1.1}$$

Nota-se trata-se de uma informação relacionada apenas ao evento analisado. Além disso, eventos de baixa probabilidade transportam mais informação.

1.2. Entropia

Definição Dada uma variável aleatória x então sua **Entropia** $H(x)$ será a quantidade média de informação recebida ao conhecer seu valor, sendo descrita pela equação abaixo:

$$H(x) = E(I(x)) = \sum_{i=1}^{N} p_i \log_2\left(\frac{1}{p_i}\right)$$

Onde $E(x)$ representa a **Esperança** da variável x, podendo ser simplificada para:

$$\boxed{H(x) = -\sum_{i=1}^{N} p_i \log_2 p_i} \tag{1.2.1}$$

Nota-se que trata-se de uma informação relacionada apenas ao processo analisado:

1. Quanto mais baixa, mais previsível;
2. Quanto mais alta, mais imprevisível;

1.3. Codificação

Definição Mapeamento **biunívoco**, cada elemento associado a um único contraelemento, entre cadeias de bits e os membros do conjunto de dados a serem condificados. Classificados em:

1. **Comprimento Fixo:** Caso todos os símbolos ocorram com a mesma probabilidade, geralmente utiliza-se este método;
   (a) `Vantagens:`
      i. Todas as folhas possuem a mesma distância da raiz;
      ii. Acesso Aleatório: Variáveis podem ser lidas em qualquer trecho da codificação;
   (b) `Entropia:` Considera-se uma variável aleatória X que assume valores entre N possibilidades equiprováveis será:

$$\boxed{H(x) = \sum_{i=1}^{N} p_i \log_2\left(\frac{1}{p_i}\right) = \sum_{i=1}^{N} \frac{1}{N} \log_2(N)} \tag{1.3.1}$$

   Desta forma, uma codificação **ótima** terá $N = 2^k$, onde $k \in \mathbb{N}$.

2. **Comprimento Variável:** Caso todos os símbolos não ocorram com a mesma probabilidade, geralmente utiliza-se este método;
   (a) `Vantagens:`
      i. Flexibilidade para se aproximar da codificação **ideal**;
      ii. Necessária para compresão de arquivos, como descrito por **Huffman**;

(b) `Entropia:` Considera-se uma variável aleatória X que assume valores entre N possibilidades equiprováveis será:

$$\boxed{H(x) = \sum_{i=1}^{N} p_i \log_2 \left(\frac{1}{p_i}\right)} \tag{1.3.2}$$

Desta forma, uma codificação **ótima** terá:

i. `Codificação Curta:` Se x_i tiver uma probabilidade alta;
ii. `Codificação Longa:` Se x_i tiver uma probabilidade baixa;

3. **Codificação Ambígua:** Organização não única dos caracteres envolvidos o que pode gerar problemas de interpretação dos dados. Deve ser **evitada**;

Será necessário evitar codificações ambíguas, pois poderá haver incerteza de informação neste caso. Desta forma, uma árvore binária deve ser criada para validar se a codificação é válida, alocando as variáveis nos terminais das ramificações.

1.4. Algoritmo de Huffman

Definição Algoritmo para construção de uma **Árvore Binária Ótima**, isto é uma codificação que possua entropia próxima a mínima necessária. Aplica-se os seguintes passos:

1. Criação de uma sub-árvore com os símbolos de **menor** probabilidade, associando-a o somatório de suas possibilidades;
2. Seleção de dois símbolos ou sub-árvores com menores probabilidades e as combine em uma nova sub-árvore;
   (a) Caso hajam símbolos ou sub-árvores com mesma probabilidade, escolha arbitrariamente;

Consequência deste algoritmo:

- Todas as codificações apresentam o mesmo comprimento esperado, logo a mesma eficiência, independente dos rótulos empregados para cada ramificação;
- Desempenhos mais próximos da entropia podem ser obtidos com sequências maiores, normalmente aplicadas em algoritmos de compressão como `LZW`;

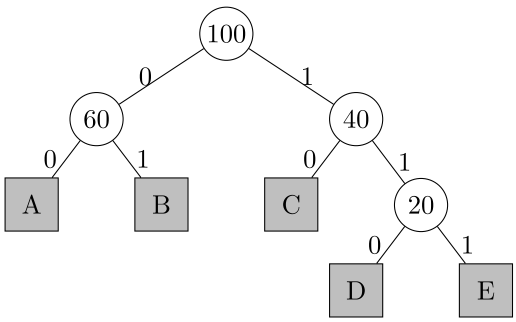


Figura 1.1: Representação da Árvore de Huffman

Considera-se como exemplo a seguinte distribuição de probabilidades:

Símbolos	Probabilidade	Codificação
A	30	00
B	30	01
C	20	10
D	10	110
E	10	111

Tabela 1: Probabilidades dos Símbolos

1.5. Distância de Hamming

Definição Representa o número de posições nos quais os dígitos correspondentes **diferem** entre si, como representado abaixo:

Original	Palavra Código
0110 0100	0100 1100

Tabela 2: Representação da Distância de Hamming

1.5.1. Deteção de `Erro de 1 bit`

Definição Criação de palavras código válidas, de modo que um erro de um **único** `bit` não produza outra palavra de código válida. Desta forma, será necessário uma codificação cuja distância de Hamming entre quaisquer palavras válidas seja de **pelo menos** 2.

Aplicação Adiciona-se um `bit` em qualquer palavra válida para que o número total de `bits 1` seja:

1. **Paridade Par:** Possui um número par de `bits 1`. Representado com `bit 0`;
2. **Paridade Ímpar:** Possui um número ímpar de `bits 1`. Representado com `bit 1`;

Generalização Considere um símbolo codificado qualquer, para **Detectar** um número E de erros será necessário uma distância mínima de Hamming $E + 1$ entre as palavras de código. Além disso, a **Correção** um número E de erros será necessário uma distância mínima de Hamming $2E + 1$.

1.5.2. Codificação de Hamming `(15, 11)`

Definição Organização de dados em `15 bits`, `11 bits` de dados e `4 bits` são redundância. Desta forma, os `bits` redundantes são suficientes para determinar a posição de qualquer erro de `1 bit` presente nos dados.

Aplicação Organize os `11 bits` de dados sequencialmente nos espaços brancos de uma matriz 4x4 como representado abaixo:

0 x	1 p	2 p	3 1
4 p	5 0	6 1	7 0
8 p	9 0	10 1	11 0
12 1	13 0	14 0	15 1

Tabela 3: Codificação de Hamming

Na sequência preenche-se os `bits de paridade`, apresentados nas posições com p, representando a paridade de cada **subgrupo** possuam como representado abaixo:

0 x	1 0	2 p	3 1
4 p	5 0	6 1	7 0
8 p	9 0	10 1	11 0
12 1	13 0	14 0	15 1

0 x	1 0	2 0	3 1
4 p	5 0	6 1	7 0
8 p	9 0	10 1	11 0
12 1	13 0	14 0	15 1

0 x	1 0	2 0	3 1
4 1	5 0	6 1	7 0
8 p	9 0	10 1	11 0
12 1	13 0	14 0	15 1

0 x	1 0	2 0	3 1
4 1	5 0	6 1	7 0
8 1	9 0	10 1	11 0
12 1	13 0	14 0	15 1

Tabela 4: Grupos de Hamming

Neste ponto pode-se detectar e localizar `erros de 1 bit`. Na sequência preenche-se o `bit de paridade do conjunto` para paridade do **grupo** como representado abaixo:

0 1	1 0	2 0	3 1
4 1	5 0	6 1	7 0
8 1	9 0	10 1	11 0
12 1	13 0	14 0	15 1

Tabela 5: Codificação Hamming Estendida

Neste ponto pode-se localizar `erros de 2 bits`.

1.5.3. Decodificação de Hamming `(15, 11)`

Definição Interpretação dos dados recebidos na **Configuração de Hamming**, analisando os `15, ou 16, bits` codificados como descrito abaixo:

1. **Transmissão Correta:**
   (a) Não houve erro nos `bits de paridade`;
   (b) Não houve erro no `bit de paridade do conjunto`;

2. **Transmissão com** `Erro de 1 bit`:
   (a) Houve erro em pelo menos um dos `bits de paridade`;
   (b) Houve erro no `bit de paridade do conjunto`;

3. **Transmissão com** `Erro de 2 bit`:
   (a) Houve erro em pelo menos um dos `bits de paridade`;
   (b) Não houve erro no `bit de paridade do conjunto`;

2. Abstração Digital

Apresentação Depois de discutido como codificar informações como sequência de bits será necessário elaborar uma forma para codifica-la fisicamente que atenda aos seguintes características:

1. **Pequeno:** Necessite de pouco espaço para armazenamento;
2. **Barato:** Economicamente acessível para produção;
3. **Estável:** Não apresentará mudanças durante seu uso;
4. **Veloz:** Fácil de acessar, transformar, combinar, transmitir e armazenar;

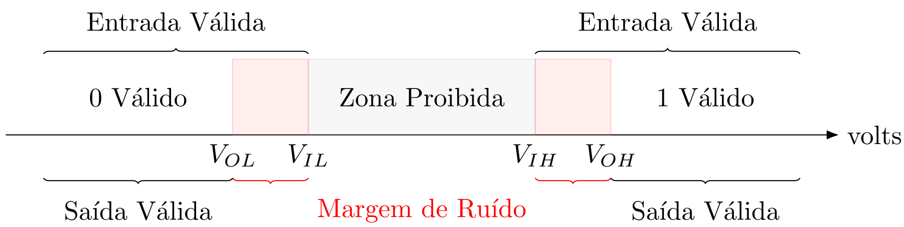
No mundo, não quântico, não é digital e são afetados por imperfeições que devem ser consideradas na descrição de modelo de conversão que consigo manter a precisão necessária para aplicação desejada.

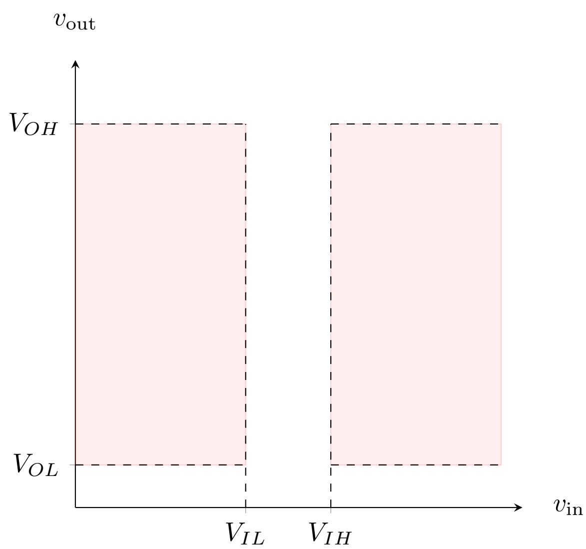
2.1. Processamento Digital

Definição Conversão, manipulação e utilização de sinais digitais para interpretação de fenômenos físicos estudados. Isso demandará algumas definições de conceitos descritas na sequência.

2.1.1. Conversão Digital

Definição Inicialmente será necessário determinar como os sinais analógicos, medições reais, serão convertido para sinais digitais para que então possam ser trabalhados, buscando métodos que atendam as condições de codificações como a representada a seguir:





2.1.2. Atraso de Propagação, t_{PD}

Definição Limitante superior para o atraso de entradas válidas para saídas válidas causado pela presença intrínseca de capacitâncias e resistências na construção dos dispositivos como representado pelo seguinte gráfico:

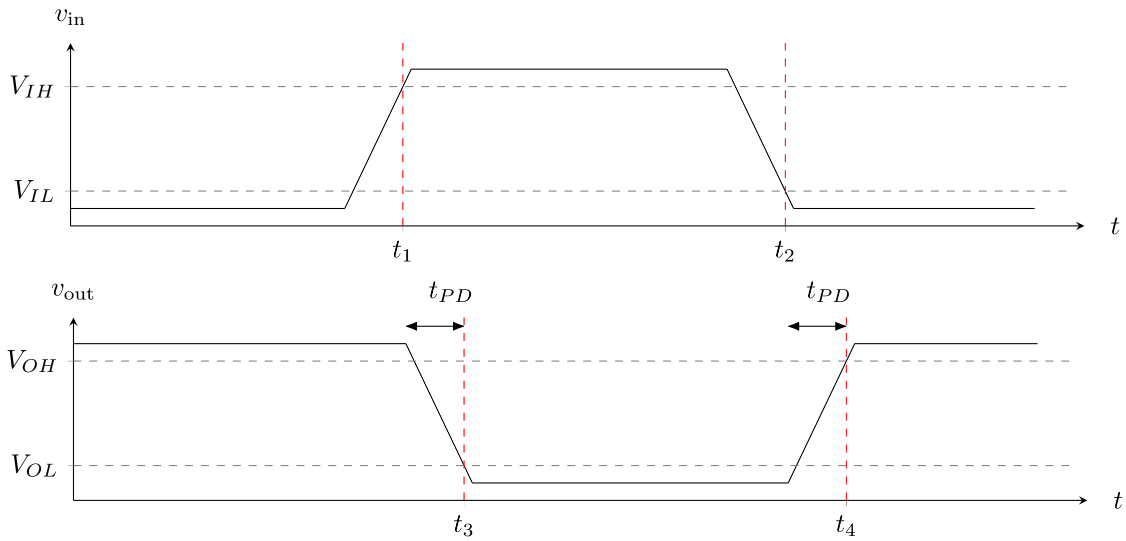


Figura 2.1: Atraso de Propagação

Note que os circuitos não serão necessariamente simétricos e portanto o t_{PD} do dispositivo será o **máximo** entre os atrasos de propagação como mostrado pela seguinte equação:

$$\boxed{t_{PD} = \max\{(t_3 - t_1), (t_4 - t_2)\}} \tag{2.1.1}$$

2.1.3. Atraso de Contaminação, t_{CD}

Definição Limitante inferior para o atraso de entradas inválidas para saídas inválidas causado pela presença intrínseca de ruídos ou interferência na atuação dos dispositivos como representado pelo seguinte gráfico:

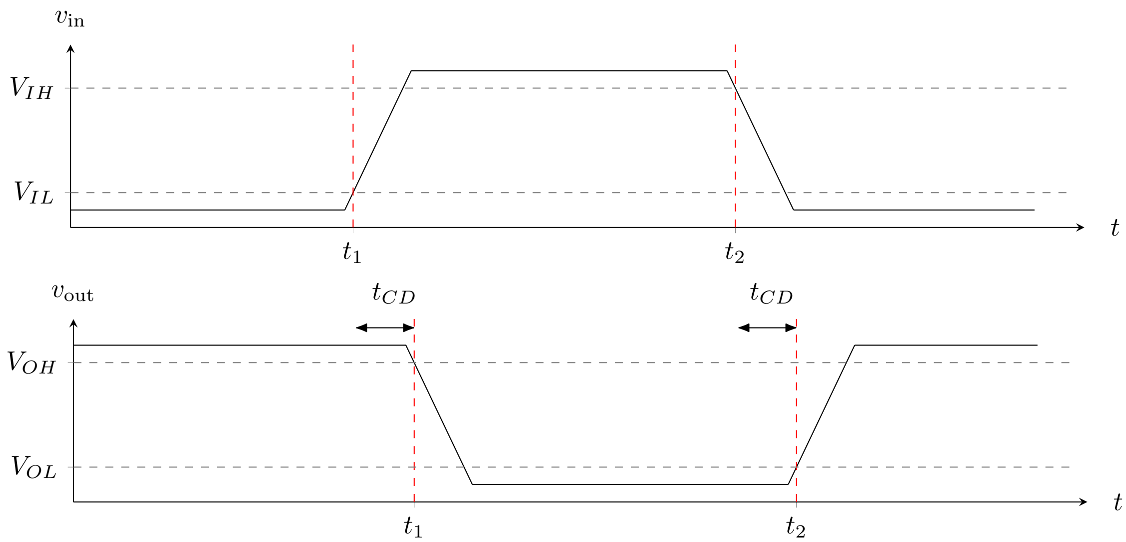


Figura 2.2: Atraso de Contaminação

Note que os circuitos não serão necessariamente simétricos e portanto o t_{CD} do dispositivo será o **mínimo** entre os atrasos de propagação como mostrado pela seguinte equação:

$$\boxed{t_{CD} = \min\{(t_3 - t_1), (t_4 - t_2)\}} \tag{2.1.2}$$

2.2. Dispositivos Combinacionais

Definição Componente eletrônico que atende as seguintes especificações:

1. **Comunicação:** Necessidades para interface com o dispositivo apresentando:
   (a) `Entradas:` Ao menos uma entrada digital;
   (b) `Saídas:` Ao menos uma saída digital;
2. **Especificação Funcional:** Qualquer saída será obtida por uma combinação possível das entradas válidas;
3. **Especificação Temporal:** Há um `Tempo de Propagação` t_{PD} mínimo necessário para que o dispositivo calcule a saída a partir de suas entradas válidas;

Além disso, um conjunto de elementos interconectados será combinacional se não viola nenhuma das seguintes regras:

1. **Condição 1:** Cada elemento individual é combinacional;
2. **Condição 2:** Cada entrada é conectada a uma, e apenas uma, saída ou fornecimento externo;
3. **Condição 3:** Não há ciclos diretos;

2.2.1. Buffer

Definição Dispositivo eletrônico que transfere tensão, mantendo sua estabilidade apesar de logicamente não alterá-lo visto que ao longo da transmisssão o acumulo de ruídos poderia modificar o sinal transmitido.

Representação Dispositivo apresentará a seguinte tabela verdade e representação em circuitos:

in	out
0	0
1	1

Tabela 6: Tabela Verdade Buffer

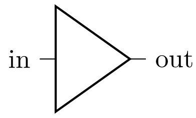
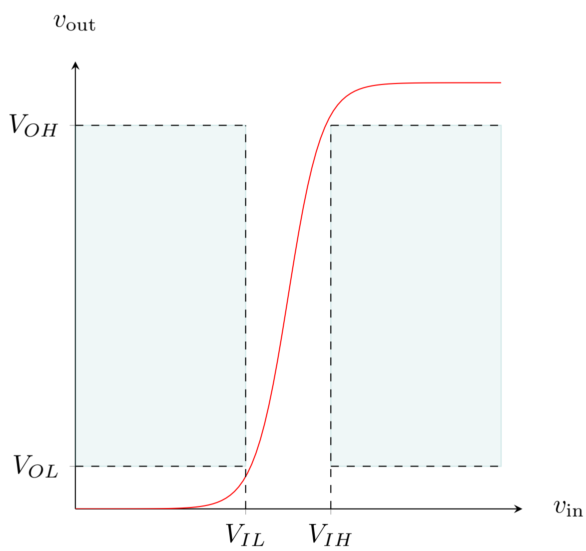


Figura 2.3: Representação Buffer

Além disso, a representação da **Curva Característica de Transferência**, relação entre a tensão de entrada com a tensão de saída, será dada pelo seguinte gráfico:



Note que este gráfico não apresenta informações dinâmicas sobre o componente, isto é, não apresenta a velocidade para se atigir o regime. Além disso, as regiões coloridas representam as **Zonas Proíbidas**, não deverá haver sinal nestas regiões.

2.3. CMOS

Definição Metodologia de projeto de circuitos digitais dominante no mercado por seu baixo consumo de energia que deve atender aos seguintes requisitos:

1. **Conexão:** Necessidades de configuração do dispositivo:
   (a) `Pull-Down:` Baixa tensão, representadas por V_{SS}, através de NMOS;
   (b) `Pull-Up:` Alta tensão, representadas por V_{DD}, através de PMOS;
2. **Dualidade:** Conexões em série PMOS são logicamente equivalentes a conexões em paralelo NMOS e vice versa;

Nota-se que os MOSFET apresentam o seguinte comportamento para suas diferentes construções:

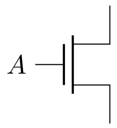


Figura 2.4: NMOS Isolado

A	NMOS	PMOS
1	1	0
0	0	1

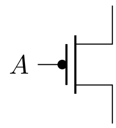
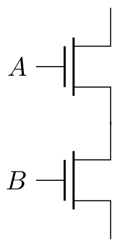
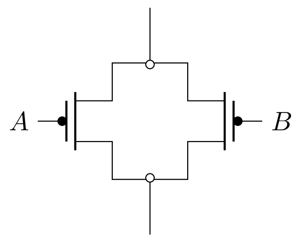


Figura 2.5: PMOS Isolado

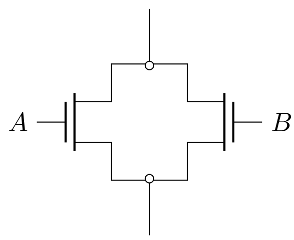
Estes dispositivos são conectados de acordo com os requisitos impostos pela construção CMOS e apresentam o seguinte comportamento para as combinações usuais:



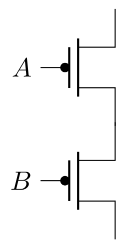
A	B	NMOS	PMOS
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0



Na sequência as combinações serão trocadas e, consequentemente, os comportamentos serão inversos como ilustrado nas tabelas verdades:



A	B	NMOS	PMOS
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



2.3.1. Inversor

Definição Dispositivo eletrônico que inverte sua tensão entrada, valores de entrada alto implicam saída baixa e valores baixos de entrada implicam saída alta.

Representação Dispositivo apresentará a seguinte tabela verdade e representação em circuitos:

in	out
0	1
1	0

Tabela 7: Tabela Verdade Inversor

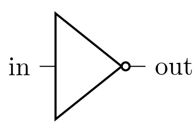


Figura 2.6: Representação Inversor

Normalmente este componente será implementado com a seguinte configuração:

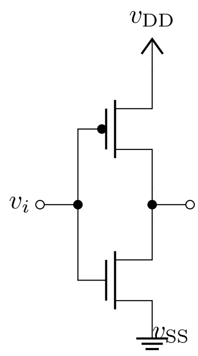
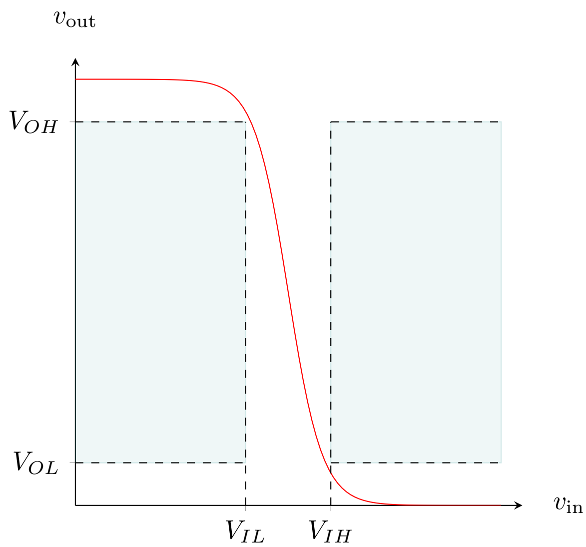


Figura 2.7: Implementação Inversor

Além disso, a representação da **Curva Característica de Transferências**, relação entre a tensão de entrada com a tensão de saída, será dada pelo seguinte gráfico:



Note que este gráfico não apresenta informações dinâmicas sobre o componente, isto é, não apresenta a velocidade para se atigir o regime. Além disso, as regiões coloridas representam as **Zonas Proíbidas**, não deverá haver sinal nestas regiões.

3. Sistemas Numéricos

Apresentação Há diferentes formas para se representar números utilizadas em situações distintas de acordo com as necessidades da aplicação, sendo os principais sistemas serão apresentadas a seguir e comparados com o **Sistema Decimal**.

3.1. Binário

Definição Sistema numérico que utiliza 2 dígitos: {0,1}, sendo que com n bits o maior número possível representável será $2^n - 1$ apresentando as seguintes posições especiais:

1. **LSB:** Least Significative Bit, bit mais à direita;
2. **MSB:** Most Significative Bit, bit mais à esquerda;

Neste sistema será possível representar números fracionários através de potências negativas de 2 que apresenta manipulação similar as entradas inteiras.

Exemplo. Representação de Números Binários:

...	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	...
	8	4	2	1	0.5	0.25	0.125	

Tabela 8: Representação de Números Binários

Desta forma, sabe-se que:

$$\boxed{1011.010_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 11.25_{10}}$$

Soma Binária Apresenta mesmo funcionamento para o sistema decimal como apresentado pelas seguintes tabelas:

Soma	`Carry-In`	Resultado	`Carry-In`
$0 + 0$	0	0	0
$0 + 1$	0	1	0
$1 + 0$	0	1	0
$1 + 1$	0	0	1
$0 + 0$	1	1	0
$0 + 1$	1	0	1
$1 + 0$	1	0	1
$1 + 1$	1	1	1

Tabela 9: Representação de Soma

Onde:

1. `Carry-In`: Representa o acúmulo de entrada da operação anterior;
2. `Carry-Out`: Representa o acúmulo de saída da operação atual;

Multiplicação Binária Apresenta funcionamento próximo para o sistema decimal apresentando uma análise mais mecânica através da técnica de `shift-and-sum`, seguindo as seguintes regras:

Exemplo. Representação de $1101_2 \times 1011_2$:

$$
\begin{array}{rl}
1101 & \\
\times\, 1011 & \\
\hline
1101 & \text{1, copia} \\
1101 & \text{1, copia} \\
0000 & \text{0, anula} \\
+\, 1101 & \text{1, copia} \\
\hline
10001111 & \text{Resultado}
\end{array}
$$

3.1.1. Conversão Decimal-Binário

Definição Há dois principais métodos:

1. **Método de Inspeção:** Decompondo os números em soma de potências de 2;
2. **Método de Divisão Sucessiva:** Realizando divisões sucessivas de 2;

Exemplo. Método de Divisão Sucessiva:

$$
\begin{array}{c|c|c|c}
8 & \underline{2} & & \\
\underline{8} & 4 & \underline{2} & \\
0 & \underline{4} & 2 & \underline{2} \\
 & 0 & \underline{2} & 1 \\
 & & 0 &
\end{array}
$$

$$\longleftarrow$$

$$8_{10} = 1000_2$$

Figura 3.1: Conversão Decimal-Binário