Circuitos Digitais

Prof. Ricardo Pedroni
ricardopedroni@utfpr.edu.br
www.rpedroni.com.br

Circuitos Combinacionais Aritméticos

Prof. Ricardo Pedroni
ricardopedroni@utfpr.edu.br
www.rpedroni.com.br

Ementa

- Lembre!
  - Sistemas Numéricos
  - Códigos binários
  - Circuitos combinacionais x Circuitos sequenciais

Ementa

- Aritmética
- Números Negativos
- Soma e Subtração binária
- Circuito Somador

Aritmética Binária

Aritmética Binária

- Como qualquer base numérica, é possível realizar operações aritméticas diretamente em números binários
- As mesmas regras da adição e subtração decimal ("vai-um", "empresta-um", etc.) valem para o binário
- Então vamos falar de outro tipo de codificação binária..

Números Negativos

Números Negativos

- Até agora: apenas números naturais e códigos de estado (binário sequencial, Gray, one-hot…)
- Existem diversos códigos para números negativos
- Os mais conhecidos são
  - Sinal-Magnitude
  - Complemento de Um
  - **Complemento de Dois**

Números Negativos

- Sinal-Magnitude
- Dedica-se o primeiro bit à esquerda (MSB) apenas para o sinal
- Se for ‘0’, é positivo, se for ‘1’, é negativo

Números Negativos

- Sinal-Magnitude

- Exemplo:
- “001” → +1
- “101” → -1
- “01101” → +13
- “11101” → -13
- “0000” → +0
- “1000” → -0

Números Negativos

- Sinal-Magnitude
- Fácil compreensão humana
- Requer mais hardware para aritmética (tenta fazer!)

Números Negativos

- Complemento de Um
- Para obter a versão negativa de um número, basta inverter todos seus bits
- O mesmo vale para obter a versão positiva

Números Negativos

- Complemento de Um

- Exemplo:
- “001” → +1
- “110” → -1
- “01101” → +13
- “10010” → -13
- “0000” → +0
- “1111” → -0

Números Negativos

- Complemento de Um

- Para fazer um cálculo
  - a – b = c
  - Inverta os bits do subtraendo (b) e conecte o "vai-um" do MSB ao "vem-um" do LSB

Números Negativos

- Complemento de Um
- (+-)Fácil compreensão humana
- Ainda requer mais hardware

Números Negativos

- Mas pera lá.

- Para melhor entender o próximo código de números negativos, é preciso entender a **adição binária**

Números Negativos

- Adição binária
- Caso mais simples: soma de 1 bit com 1 bit
- Duas entradas (1 bit cada), saída (soma) e o *carry* ("vai-um")

Números Negativos

- Adição binária

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 0 (com carry)

- E o "vai-um" (conhecido como *carry*)?

Números Negativos

- Adição binária

- Soma

a	b	Soma
0	0	0
0	1	1
1	0	1
1	1	0

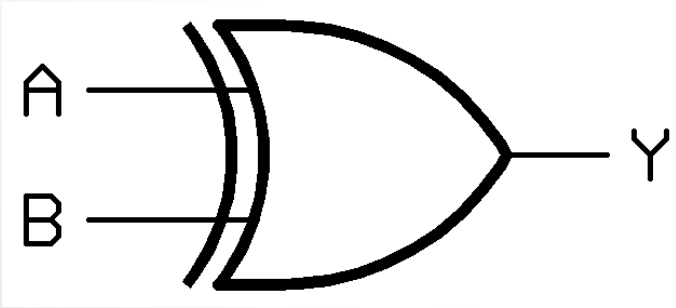
Carry

a	b	Carry
0	0	0
0	1	0
1	0	0
1	1	1

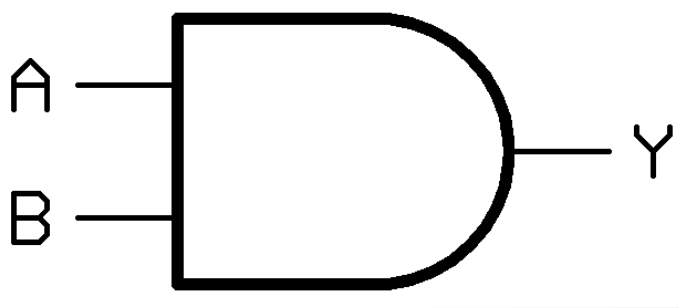
Números Negativos

- Adição binária

- Soma



Carry



Números Negativos

- Ok! Voltando..
- Código Complemento de Dois

Números Negativos

- Antes de mais nada..
  - Complemento de 9
  - Complemento de 10

Números Negativos

- Código Complemento de Dois
- Para fazer um cálculo
  - a – b = c
  - Inverta os bits do subtraendo (b) e coloque em '1' o "entra-um" do LSB.
- Basta inverter o valor positivo e somar '1' para ter seu valor negativo (e vice versa!)
- Hardware simples

Números Negativos

- Código Complemento de Dois

- Exemplo:
- “00000100” → +4
- “11111100” → -4
- “001011” → +11
- “110101” → -11
- “10000000” → -128

- IMPORTANTE: MSB ‘1’ → Negativo, MSB ‘0’ → Positivo

Números Negativos

- Código Complemento de Dois

- Abrange a faixa de

$$-2^{N-1} \leq x \leq 2^{N-1} - 1$$

- Agora vamos voltar a aritmética..

Adição Sem Sinal

Adição Sem Sinal

- Caso 1: Entradas N bits, Saída N bits

Adição Sem Sinal

- Caso 2: Entradas N bits, Saída N+1 bits

- Exemplo

Adição e Subtração Com Sinal

+ e - Com Sinal

- A subtração é similar à adição
- Invés de *carry* existe o *borrow* ("empresta-um")

+ e - Com Sinal

- A subtração computacional é feita através do complemento de dois
- Números negativos são representados por complemento de dois
- Para subtrair um número de outro, basta somar sua versão negativa com o outro número
  - a – b = a + (-b)
- Conclusão:
- Basta sempre usarmos **somadores**!

+ e - Com Sinal

- Porém, ao usar um somador, qualquer valor negativo precisa estar em seu complemento de dois
- Exemplo:
  - 6 – 4 = “0110” – “0100” = “0110” + “1100” = 6 + (-4)

+ e - Com Sinal

- A soma com complemento de dois funciona corretamente desde que o resultado caia na faixa

$$-2^{N-1} \leq x \leq 2^{N-1} - 1$$

+ e - Com Sinal

- Se os dois operandos possuem sinais contrários, certamente a soma estará correta
- Se os dois últimos carries são iguais, a soma está correta
- Se os dois últimos carries são diferentes, temos um overflow – pode ser corrigido usando o último carry

+ e - Com Sinal

- Uma forma de garantir a operação em complemento de dois correta é estender o sinal dos operandos para N+1 bits, assim garantindo que não haverá overflow

- Pode-se truncar o valor para N bits, caso o bit descartado é apenas uma extensão do sinal

Circuitos Aritméticos

Circuitos Aritméticos

- Basicamente dois tipos
  - Meio somador (Half adder)
  - Somador completo (Full adder)
- Diferem no uso do *carry* (tanto in quanto out)
- Lembre que tanto soma quanto subtração usam somadores (subtração usa o completemento de dois)

Circuitos Aritméticos

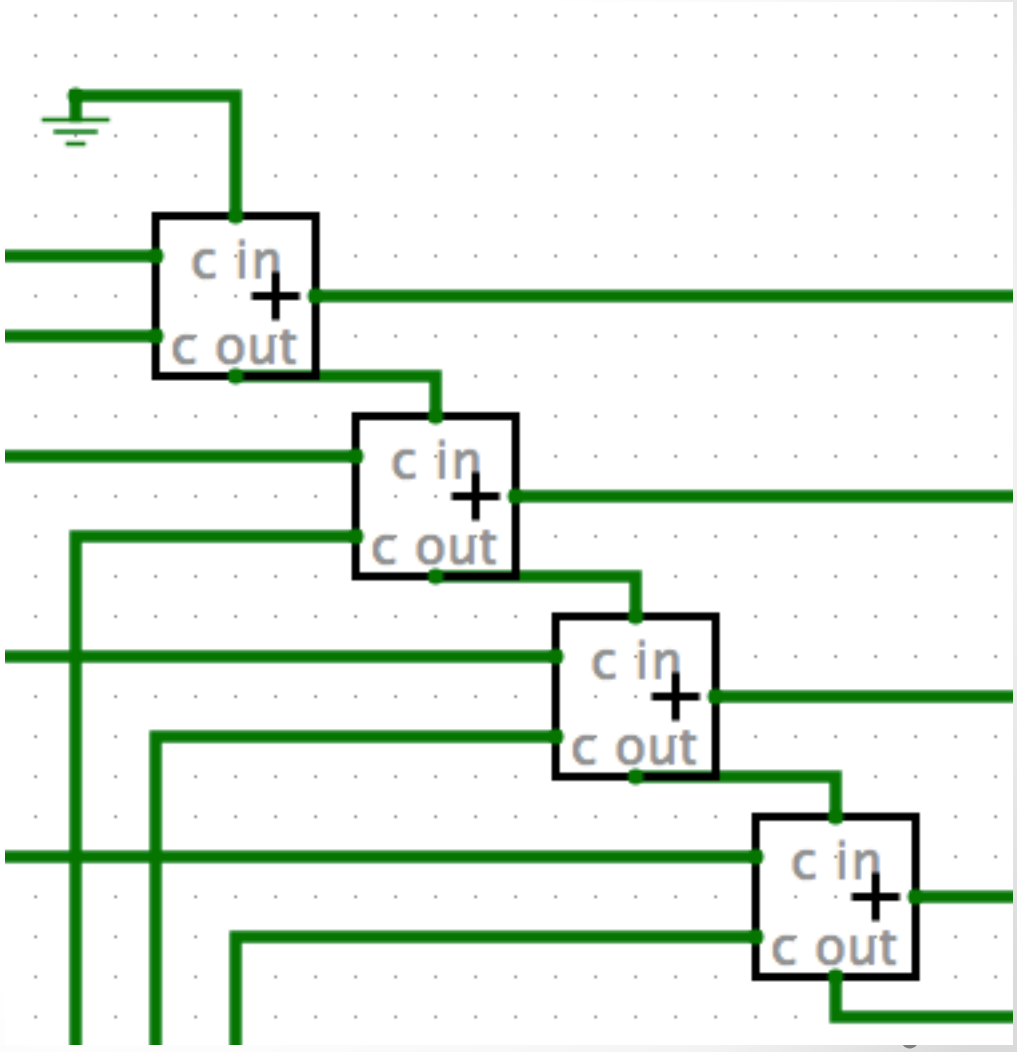
- Half adder
  - É um somador com o *carry out*
  - Não considera o *carry in*

Circuitos Aritméticos

- Full adder
  - Mesma coisa que o half adder porém utiliza o *carry in*
  - Pode ser "cascateado" (i.e. vários full adders podem ser conectados em sequência para criar um somador de N bits)

Circuitos Aritméticos

- Full adder
- Conhecido como carry ripple
- Cascateia vários full adders em sequência



Circuitos Aritméticos

- Exercício
  - Construa seu próprio full adder, utilizando apenas portas lógicas
  - Usando um full adder de 4 bits (com acesso aos carries), construa um circuito que tenha 5 bits de saída, contendo o valor correto de qualquer soma possível.