

4 SISTEMAS DE NUMERAÇÃO RESIDUAL

Metas:

Estudar uma maneira de codificar grandes números como uma coleção de números menores para simplificar e acelerar algumas operações

Destaques do capítulo:

Conjuntos de módulos, gama, operações aritméticas

Muitos conjuntos de módulos possíveis: *tradeoffs*

Conversões entre RNS e binário

Teorema chinês do resto

Por que as aplicações de RNS são limitadas?

Computer Arithmetic, Number Representation

SLIDE 1

Neste capítulo vamos aprender um método muito usado para codificar números muito grandes e simplificar operações aritméticas. Este sistema de numeração, chamado sistema de numeração residual é baseado em resíduos associados a uma determinada base ou bases (residue number systems RNS).

Vamos definir conceitos como conjunto de modulos, faxia dinâmica, conversões de binário a resíduo, resíduo a binário por meio do teorema do resto chinês, assim como as aplicações que este novo sistema de numeração tem.

4 SISTEMAS DE NUMERAÇÃO RESIDUAL

- 4.1 Fundamentação RNS.**
- 4.2 Codificação e Decodificação em RNS.**
 - 4.2.1 Conversão Binário para RNS.**
 - 4.2.2 Conversão RNS para Binário.**
- 4.3 Aritmética em RNS.**
- 4.4 Escolhendo o Módulo RNS.**
- 4.5 Implementação em hardware.**

Computer Arithmetic, Number Representation

SLIDE 2

Primeiro vamos definir a representação RNS e a aritmética associada a ela. Em seguida vamos escolher os conjunto de módulos (ou bases) mais apropriadas para um sistema otimizado. O terceiro passo consiste na codificação e decodificação dos números residuais.

Por fim, abordaremos as operações mais complicadas com o intuito de determinar quais serão as aplicações mais apropriadas.

4 SISTEMAS DE NUMERAÇÃO RESIDUAL

4.1 Fundamentação RNS.

4.2 Codificação e Decodificação em RNS.

4.2.1 Conversão Binário para RNS.

4.2.2 Conversão RNS para Binário.

4.3 Aritmética em RNS.

4.4 Escolhendo o Módulo RNS.

4.5 Implementação em hardware.

Computer Arithmetic, Number Representation

SLIDE 3

Desta forma, iniciando pela definição de RNS e a sua aritmética associada.

4.1. FUNDAMENTAÇÃO – RNS

No Sistema Numérico de Resíduos um número binário é convertido em paralelo para um conjunto de **resíduos** correspondente aos restos da divisão por um conjunto de **módulos**.

$$M = \{m_1, m_2, \dots, m_n\}$$

Os módulos devem ser primos entre si para permitir a conversão do valor residual para uma solução binária final.

$$\text{MDC}(m_i, m_j) = 1 \quad , \quad \text{onde } i \neq j$$

A faixa dinâmica (DR) é dada por:

$$DR = m_1 \times m_2 \times \dots \times m_n$$

Assim:

$$\begin{array}{l} X = (R_1, R_2, \dots, R_n) \\ \text{Binaria} \quad \quad \quad \text{Resíduos para os módulos } m_i \\ R_i = X \bmod m_i = |X|_{m_i} \quad , \quad 0 \leq R_i < m_i \end{array}$$

Computer Arithmetic, Number Representation

SLIDE 4

A formulação do sistema numérico residual é proveniente de um verso chinês que faz a seguinte pergunta:

Qual o número que possui resto igual a 2,3,2 quando dividido pelos números 7,5,3 respectivamente? Este verso nos pede, essencialmente, para converter a representação (2,3,2) codificada no sistema numérico residual (baseado nos módulos (7,5,3)) em um formato decimal padrão.

Nesta formulação chamaríamos (2,3,2) de resíduos e (7,5,3) de módulos.

Vejamos então as formulações no Slide.

(OBS:) A faixa dinâmica seria o produto dos módulos.

4.1. FUNDAMENTAÇÃO – RNS

EXEMPLO 4.1: Para $M = \{5, 8, 11\}$, as representações de 32_{10} e 48_{10} seriam:

$$\begin{aligned}|32|_5 &= 2, & |32|_8 &= 0, & |32|_{11} &= 10 \\|48|_5 &= 3, & |48|_8 &= 0, & |48|_{11} &= 4.\end{aligned}$$

A soma de 32_{10} e 48_{10} , utilizando RNS, seria:

$$|2+3|_5 = |0|_5, \quad |0+0|_8 = |0|_8, \quad |10+4|_{11} = |3|_{11}$$

Fazendo a verificação ($32_{10} + 48_{10} = 80_{10}$):

$$|80|_5 = 0, \quad |80|_8 = 0, \quad |80|_{11} = 3$$

O intervalo de representação (DR) é a multiplicação dos módulos
 $5 \times 8 \times 11 = 440$ ou $[0, DR-1] = [0, 439]$ *

* O DR também pode ser representado pelo número de bits da multiplicação.

Computer Arithmetic, Number Representation

SLIDE 5

Acompanhemos o texto observando o slide acima.

No exemplo acima, temos os números 32(base 10) e 48(base 10) e os módulos (5,8,11).

Para achar as representações desses números em RNS devemos achar o resto da divisão, por exemplo, do número 32 dividido por 5 o resto da divisão é igual a 2, assim sendo $|32|_5 = 2$. Vejamos no Slide.

Dois seria a representação em RNS de (32 módulo 5).

Da mesma forma teríamos o resultado (0) para (32 módulo 8) e um resultado igual a (3) para (48 módulo 5). E assim por diante.

Se somarmos os resíduos teremos: (lembrando que resíduos são os restos das divisões pelos respectivos módulos)

A soma de $2+3$ seria igual a 5 (onde 2 e 3 são os restos da divisão de 32 e 48 pelo módulo 5). Entretanto após a soma devemos aplicar novamente o módulo 5 ao resultado. Então a soma de $(2+3) = 5$ ao aplicar o módulo 5 (extrair novamente o resto da divisão por 5) seria igual a zero.

Vejamos então a prova do funcionamento desta Aritmética. Se somarmos os números 32(base 10) e 48(base 10) teremos o resultado igual a 80(base 10).

Agora, se observarmos as somas feitas anteriormente teremos que 80 (aplicado aos respectivos módulos) será igual aos resultados obtidos nestas somas feitas em RNS.

Neste exemplo teremos a faixa de representação (DR) obtido pela multiplicação dos módulos, como demonstrado no slide.

4.1. FUNDAMENTAÇÃO – RNS

O produto M do módulo k relativamente primo é o intervalo dinâmico

$$M = m_{k-1} \times \dots \times m_1 \times m_0$$

$$\text{Para RNS}(8 | 7 | 5 | 3), \quad M = 8 \times 7 \times 5 \times 3 = 840$$

Números negativos: complemento relativo a M

$$\langle -x \rangle_{m_j} = \langle M - x \rangle_{m_j}$$

$$\begin{array}{rcl} 1 & = & (1 | 1 | 1 | 1)_{\text{RNS}} \\ -1 & = & (840 - 1 | 840 - 1 | 840 - 1 | 840 - 1)_{\text{RNS}} = (7 | 6 | 4 | 2)_{\text{RNS}} \end{array}$$

Here are some example numbers in our default RNS(8 | 7 | 5 | 3):

$$(0 | 0 | 0 | 0)_{\text{RNS}} \quad \text{Representa } 0 \text{ ou } 840 \text{ ou } -840 \dots$$

$$(1 | 1 | 1 | 1)_{\text{RNS}} \quad \text{Representa } 1 \text{ ou } 841 \text{ ou } -839 \dots$$

$$(2 | 2 | 2 | 2)_{\text{RNS}} \quad \text{Representa } 2 \text{ ou } 842 \text{ ou } -838 \dots$$

$$(0 | 1 | 3 | 2)_{\text{RNS}} \quad \text{Representa } 8 \text{ ou } 848 \text{ ou } -832 \dots$$

$$(5 | 0 | 1 | 0)_{\text{RNS}} \quad \text{Representa } 21 \text{ ou } 861 \text{ ou } -819 \dots$$

$$(0 | 1 | 4 | 1)_{\text{RNS}} \quad \text{Representa } 64 \text{ ou } 904 \text{ ou } -776 \dots$$

$$(2 | 0 | 0 | 2)_{\text{RNS}} \quad \text{Representa } 770 \text{ ou } 1610 \text{ ou } -70 \dots$$

$$(7 | 6 | 4 | 2)_{\text{RNS}} \quad \text{Representa } 839 \text{ ou } 1679 \text{ ou } -1 \dots$$

Podemos considerar o intervalo de RNS (8 | 7 | 5 | 3) como [-420, 419] ou qualquer outro conjunto de 840 inteiros consecutivos

SLIDE 6

Computer Arithmetic, Number Representation

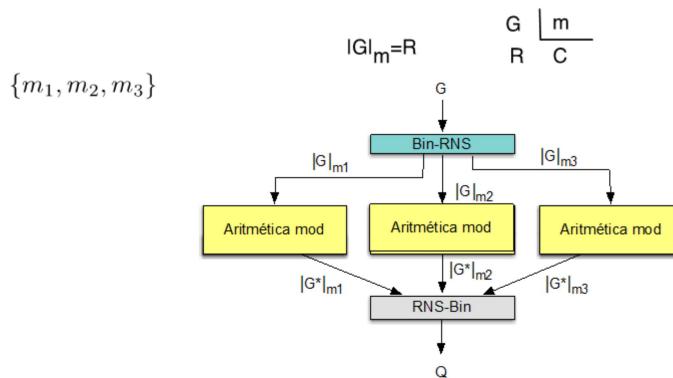
A faixa de valores que pode ser representada em RNS é determinada diretamente pelo produto dos módulos.

Por exemplo. Tomemos o conjunto de módulo (8,7,5,3), como apresentado no slide. Observamos que a faixa dinâmica será igual a 840. Estes 840 possíveis valores podem ser utilizados, por exemplo, para representar números de 0 a 839, de -420 a +419, ou qualquer intervalo de 840 inteiros.

Observação: Números negativos são representados utilizando um sistema de complemento.

4.1. FUNDAMENTAÇÃO – RNS

Nos Sistemas de Números de Resíduos, um número binário é convertido em paralelo para um conjunto de palavras residuais correspondentes aos restos de valores de módulos:



Computer Arithmetic, Number Representation

SLIDE 7

Modelo de implementação para um sistema em RNS.

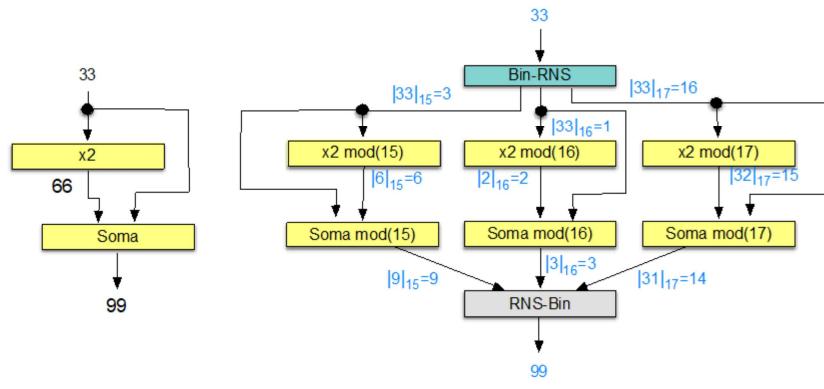
Para realizar cálculos no sistema RNS, na prática, devemos transformar o número binário, com o qual deseja-se realizar o cálculo, em um conjunto de palavras residuais, correspondentes aos restos de valores de módulos.

Consequentemente utilizamos unidades aritméticas para realizar operações modulares (já no sistema RNS) e finalmente convertemos os resultados de volta para o sistema binário. Resumindo passamos de Binário para RNS, realizamos as operações aritméticas e voltamos de RNS para Binário.

4.1. FUNDAMENTAÇÃO – RNS

Para o conjunto de módulos $\{15, 16, 17\} = \{2^4-1, 2^4, 2^4+1\}$ e uma entrada $G=33$, a solução para a operação $Q=33*2+33=99$.

A solução binária requer grandes multiplicadores e somadores em comparação com a solução RNS.



Computer Arithmetic, Number Representation

SLIDE 8

Dependendo da operação que desejamos realizar, teremos a implementação do sistema em RNS, menor e mais eficiente quando comparado à implementação em sistema binário comum.

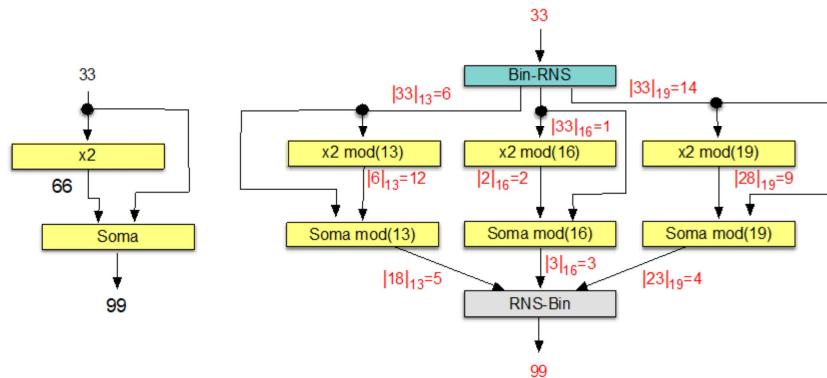
Vejamos o exemplo no diagrama para a operação ($Q=33*2+33=99$) utilizando os módulos $(15, 16, 17)$.

Observe que os valores modulares podem ser realizados em paralelo.

4.1. FUNDAMENTAÇÃO – RNS

Para o conjunto de módulos $\{13, 16, 19\} = \{2^4 - 3, 2^4, 2^4 + 3\}$ e uma entrada $G = 33$, a solução para a operação $Q = 33 * 2 + 33 = 99$.

A solução binária requer grandes multiplicadores e somadores em comparação com a solução RNS.



Computer Arithmetic, Number Representation

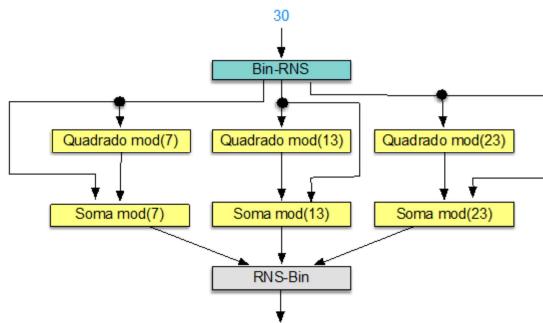
SLIDE 9

Aqui a solução utilizando um outro conjunto de módulos $(13, 16, 19)$.

PROBLEMAS

Problema 4.1. Obtenha as faixas dinâmicas para os seguintes conjuntos de módulos: a) $M_1 = \{3, 5, 7, 17\}$; b) $M_2 = \{15, 16, 17\}$; c) $M_3 = \{7, 13, 23\}$.

Problema 4.2. Indique os valores de saída para todos os blocos. A saída final está na faixa dinâmica permitida para o modulo?



SLIDE 10

Gabarito no Moodle.

4 SISTEMAS DE NUMERAÇÃO RESIDUAL

4.1 Fundamentação RNS.

4.2 Codificação e Decodificação em RNS.

 4.2.1 Conversão Binário para RNS.

 4.2.2 Conversão RNS para Binário.

4.3 Aritmética em RNS.

4.4 Escolhendo o Módulo RNS.

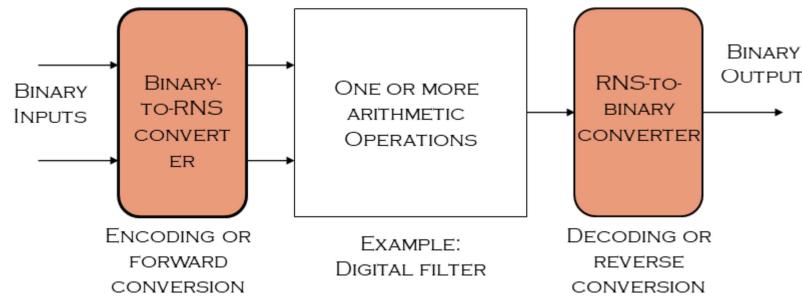
4.5 Implementação em hardware.

Computer Arithmetic, Number Representation

SLIDE 11

Agora veremos como realizar a codificação e decodificação dos números residuais.

4.2 CODIFICAÇÃO E DECODIFICAÇÃO EM RNS



Quanto mais a quantidade de computação executada entre o conversão inicial e conversão inversa final (reconversão), maiores são os benefícios da representação RNS.

Computer Arithmetic, Number Representation

SLIDE 12

Para realizar cálculos no sistema RNS seguimos o fluxo transformando entradas binárias para representações em RNS. Logo após realizamos as operações aritméticas desejadas e por fim convertemos os resultados do sistema em RNS para binário.

Observação: Temos o bloco de conversão à direta (Binário para RNS) como um bloco relativamente simples de implementar. Já a conversão reversa (RNS para Binário) possui uma maior complexidade.

4.2.1 CONVERSÃO BINARIA-RNS

EXEMPLO 4.2: Represente o número $y = (1010\ 0100)_2 = (164)_{10}$ em RNS $(8 | 7 | 5 | 3)$

Obtemos $y = 2^7 + 2^5 + 2^2$;

$$\begin{aligned}x_3 &= \langle y \rangle_8 = \langle 0 + 0 + 4 \rangle_8 = 4 \\x_2 &= \langle y \rangle_7 = \langle 2 + 4 + 4 \rangle_7 = 3 \\x_1 &= \langle y \rangle_5 = \langle 3 + 2 + 4 \rangle_5 = 4 \\x_0 &= \langle y \rangle_3 = \langle 2 + 2 + 1 \rangle_3 = 2\end{aligned}$$

TABELA 4.1 RESÍDUOS DAS 10 PRIMEIRAS POTÊNCIAS DE 2

| I | 2^I | $\langle 2 \rangle_8$ | $\langle 2 \rangle_7$ | $\langle 2 \rangle_5$ | $\langle 2 \rangle_3$ |
|-----|-------|-----------------------|-----------------------|-----------------------|-----------------------|
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 2 | 2 | 2 | 2 | 2 |
| 2 | 4 | 4 | 4 | 4 | 1 |
| 3 | 8 | 0 | 1 | 3 | 2 |
| 4 | 16 | 0 | 2 | 1 | 1 |
| 5 | 32 | 0 | 4 | 2 | 2 |
| 6 | 64 | 0 | 1 | 4 | 1 |
| 7 | 128 | 0 | 2 | 3 | 2 |
| 8 | 256 | 0 | 4 | 1 | 1 |
| 9 | 512 | 0 | 1 | 2 | 2 |

Computer Arithmetic, Number Representation

SLIDE 13

Dado o exemplo acima vejamos:

A Tabela 4.1 mostra como converter números binários para RNS (módulos: $8 | 7 | 5 | 3$).

Observemos que o número 164 pode ser decomposto pela soma: $y = 2^7 + 2^5 + 2^2 = (128 + 32 + 4) = 164$

Podemos utilizar essa decomposição para facilitar nosso cálculo. Por exemplo:

Diremos que X_0 é referente à operação $164 \bmod 3$. Por exemplo, pela decomposição $y = 2^7 + 2^5 + 2^2$ podemos inferir que $128 \bmod 3$ é igual a 2, bem como $32 \bmod 3$ também é igual a 2. Já $4 \bmod 3$ é igual a 1. Então:

$$x_0 = \langle y \rangle_3 = (128 \bmod 3) + (32 \bmod 3) + (4 \bmod 3) = \langle 2 + 2 + 1 \rangle_3 = (5 \bmod 3) = 2$$

De forma similar, teremos as seguintes resoluções para os demais módulos:

$$x_1 = \langle y \rangle_5 = (128 \bmod 5) + (32 \bmod 5) + (4 \bmod 5) = \langle 3 + 2 + 4 \rangle_5 = (9 \bmod 5) = 4$$

$$x_2 = \langle y \rangle_7 = (128 \bmod 7) + (32 \bmod 7) + (4 \bmod 7) = \langle 2 + 4 + 4 \rangle_7 = (10 \bmod 7) = 3$$

$$x_3 = \langle y \rangle_8 = (128 \bmod 8) + (32 \bmod 8) + (4 \bmod 8) = \langle 4 + 0 + 0 \rangle_8 = (4 \bmod 8) = 4$$

4.2.1 CONVERSÃO BINARIA-RNS

EXEMPLO 4.3: Represente o número $y = (1010\ 0100)_2 = (164)_{10}$ em RNS($15 \mid 16 \mid 17$)

Obtemos $y = 2^7 + 2^5 + 2^2$;

$$x_3 = \langle y \rangle_{15} = \langle 8 + 2 + 4 \rangle_{15} = 14$$

$$x_2 = \langle y \rangle_{16} = \langle 0 + 0 + 4 \rangle_{16} = 4$$

$$x_1 = \langle y \rangle_{17} = \langle 9 + 15 + 4 \rangle_{17} = 11$$

TABELA 4.2 RESIUDOS DAS 10 PRIMEIRAS POTENCIAS 2

| I | 2^I | $\langle 2^I \rangle_{15}$ | $\langle 2^I \rangle_{16}$ | $\langle 2^I \rangle_{17}$ |
|-----|-------|----------------------------|----------------------------|----------------------------|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 2 | 2 | 2 | 2 |
| 2 | 4 | 4 | 4 | 4 |
| 3 | 8 | 8 | 8 | 8 |
| 4 | 16 | 1 | 0 | 16 |
| 5 | 32 | 2 | 0 | 15 |
| 6 | 64 | 4 | 0 | 13 |
| 7 | 128 | 8 | 0 | 9 |
| 8 | 256 | 1 | 0 | 1 |
| 9 | 512 | 2 | 0 | 2 |

Computer Arithmetic, Number Representation

SLIDE 14

De forma similar, vemos a solução para um conjunto de módulos diferentes ($15, 16, 17$):

4.2.2 CONVERSÃO RNS-BINARIO

Quebra-cabeça, devido ao estudioso chinês Sun Tzu, há mais de 1500 anos

Que número tem os restos de 2, 3 e 2?
quando dividido por 7, 5 e 3, respectivamente?

$$X = (2 | 3 | 2)_{RNS(7|5|3)} = (?)_{TEN}$$

Computer Arithmetic, Number Representation

SLIDE 15

Para realizar a conversão reversa lembremos daquilo que foi apresentado em comentário em alguns slides anteriormente.

OBS: Solução deste problema seria igual a 23.

4.2.2 CONVERSÃO RNS-BINARIO

A representação RNS pode ser convertida de volta para binário (X) usando:

a) Teorema Chinês do Resto (CRT):

$$X = \left| \sum_{i=1}^n \hat{m}_i |\hat{m}_i^{-1}|_{m_i} \times R_i \right|_M , \text{ onde } M = \prod_{i=1}^n m_i$$

$$\hat{m}_i = M/m_i$$

$$|\hat{m}_i^{-1}|_{m_i} \hat{m}_i |_{m_i} = 1$$

$|\hat{m}_i^{-1}|_{m_i}$ represents the multiplicative inverse of \hat{m}_i with respect to modulus m_i

b) Novo CRT-I:

$$X = \left| \sum_{i=1}^n |V_i R_i|_{\hat{m}_1} \right|_{\hat{m}_1} m_1 + R_1 , \text{ onde } V_1 = \frac{|\hat{m}_1^{-1}|_{m_1} \hat{m}_1 - 1}{m_1}$$

$$V_i = |\hat{m}_i^{-1}|_{m_i} \frac{\hat{m}_i}{m_1} \quad \text{for} \quad 2 \leq i \leq n$$

SLIDE 16

Na prática, utilizamos alguns algoritmos para realizar a conversão reversa. Acima apresentam-se dois algoritmos. O CRT e o novo CRT.

Vejamos um exemplo utilizando o CRT.

Consideremos os resíduos e seus respectivos módulos (R_1, R_2, R_3) = (3(mod5), 1(mod7), 6(mod8)). Primeiro calcularemos M , que consiste na multiplicação entre todos os módulos $M = (5 \times 7 \times 8) = 280$

Em seguida calcularemos \hat{m}_i , que consiste, na prática, na multiplicação entre dois módulos $\hat{m}_1 = M/m_1 = 280/5 = 56$, $\hat{m}_2 = 280/7 = 40$, $\hat{m}_3 = 280/8 = 35$. E as multiplicativas inversas que podem ser obtidas a partir de <https://planetcalc.com/3311/>. Multiplicativa inversa de \hat{m}_i : $\text{mod}(\hat{m}_1, m_1) = 1$, $\text{mod}(\hat{m}_2, m_2) = 3$, e $\text{mod}(\hat{m}_3, m_3) = 3$.

Aplicando o algoritmo CRT e aplicaremos ao módulo M :

$$X = \text{Mod}(56 * 1 * R_1 + 40 * 3 * R_2 + 35 * 3 * R_3, 280)$$

Substituindo os valores de resíduo inicial (R_1, R_2, R_3) = (3, 1, 6), obtemos $X = 78$.

Podemos então retirar a prova:

$$78 \pmod{5} = 3 \pmod{5}$$

$$78 \pmod{7} = 1 \pmod{7}$$

$$78 \pmod{8} = 6 \pmod{8}$$

Mostrando que está correta a conversão.

4.2.2 CONVERSÃO RNS-BINARIO

EXEMPLO 4.4: $\{m_1, m_2, m_3\} = \{16, 15, 17\}$ e $\{R_1, R_2, R_3\} = \{15, 14, 16\}$:

a) Teorema Chinês do Resto (CRT):

$$X = \left| \sum_{i=1}^n \hat{m}_i |\hat{m}_i^{-1}|_{m_i} \times R_i \right|_M , \text{ onde } \begin{aligned} M &= \prod_{i=1}^n m_i \\ \hat{m}_i &= M/m_i \\ \left| |\hat{m}_i^{-1}|_{m_i} \hat{m}_i \right|_{m_i} &= 1 \end{aligned}$$

b) Novo CRT-I:

$$X = \left| \sum_{i=1}^n |V_i R_i|_{\hat{m}_1} \right|_{\hat{m}_1} m_1 + R_1 , \text{ onde } \begin{aligned} V_1 &= \frac{|\hat{m}_1^{-1}|_{m_1} \hat{m}_1 - 1}{m_1} \\ V_i &= |\hat{m}_i^{-1}|_{m_i} \frac{\hat{m}_i}{m_1} \quad \text{for} \quad 2 \leq i \leq n \end{aligned}$$

SLIDE 17

Vejamos outro exemplo:

1º calculamos $M = 16 \times 15 \times 17 = 4080$

2º calculamos $\hat{m}_i = (255, 272, 240)$

3º retiramos sua multiplicativa inversa: (15, 8, 9)

(<https://planetcalc.com/3311/>)

Por fim multiplicamos todos esses resultados, os somamos e aplicamos ao módulo M encontrado no primeiro passo:

Multiplicação dos resultados: (57375, 30464, 34560)

Sua soma: (122399)

$X = 4079$

Para verificar o resultado pode-se utilizar: <https://www.dcode.fr/chinese-remainder>

PROBLEMAS

Problema 4.3. Represente o número $y = 1010\ 0100_2 = 200_{10}$ para os seguintes conjuntos de módulos: a) $M1=\{3,5,7,17\}$; b) $M2=\{16,15,17\}$; c) $M3=\{7,13,23\}$.

Problema 4.4. Obtenha o valor de saída aplicando a equação CRT para os seguintes conjuntos de módulos:

- a) $\{m_1,m_2,m_3,m_4\}=\{3,5,7,17\}$ e $\{R1,R2,R3,R4\}=\{2, 0, 4, 13\}$;
- b) $\{m_1,m_2,m_3\}=\{16,15,17\}$ e $\{R1,R2,R3\}=\{8, 5,13\}$;
- c) $\{m_1,m_2,m_3\}=\{7,13, 23\}$ e $\{R1,R2,R3\}=\{4, 5,16\}$.

Problema 4.5. Obtenha o valor de saída aplicando a equação Novo CRT-I para os módulos apresentados no exemplo anterior.

SLIDE 18

Gabarito no Moodle

4 SISTEMAS DE NUMERAÇÃO RESIDUAL

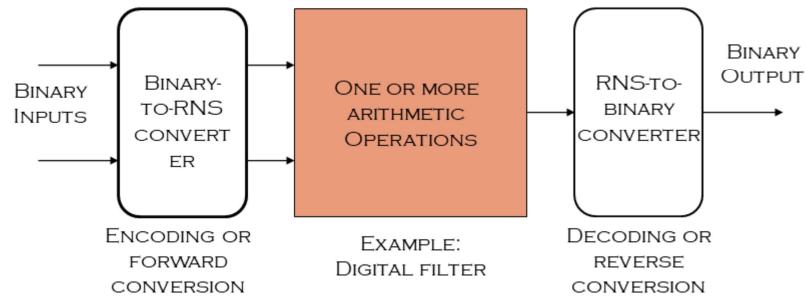
- 4.1 Fundamentação RNS.
- 4.2 Codificação e Decodificação em RNS.
 - 4.2.1 Conversão Binário para RNS.
 - 4.2.2 Conversão RNS para Binário.
- 4.3 Aritmética em RNS.
- 4.4 Escolhendo o Módulo RNS.
- 4.5 Implementação em hardware.

Computer Arithmetic, Number Representation

SLIDE 19

Vejamos agora o assunto relacionado à aritmética em RNS.

4.3 ARITMÉTICA EM RNS



Quanto mais a quantidade de computação executada entre o conversão inicial e conversão inversa final (reconversão), maiores são os benefícios da representação RNS.

Computer Arithmetic, Number Representation

SLIDE 20

Lembremos novamente a representação do sistema aritmético RNS. Veremos agora o bloco de operações aritméticas.

4.3 ARITMÉTICA EM RNS

$Y = (0100\ 0101)_2 = (69)_{10}$ em RNS($8 | 7 | 5 | 3$) e
 $Z = (0000\ 1100)_2 = (12)_{10}$ em RNS($8 | 7 | 5 | 3$), são
 $(5 | 6 | 4 | 0)_{RNS(8|7|5|3)}$ e $(4 | 5 | 2 | 0)_{RNS(8|7|5|3)}$

Multiplicação de Y e Z:

$$\langle 5 \times 4 \rangle_8 = 4; \langle 6 \times 5 \rangle_7 = 2; \langle 4 \times 2 \rangle_5 = 3; \langle 0 \times 0 \rangle_3 = 0$$

$$(4 | 2 | 3 | 0)_{RNS(8|7|5|3)}$$

Soma:

$$\langle 5 + 4 \rangle_8 = 1; \langle 6 + 5 \rangle_7 = 4; \langle 4 + 2 \rangle_5 = 1; \langle 0 + 0 \rangle_3 = 0$$

$$(1 | 4 | 1 | 0)_{RNS(8|7|5|3)}$$

Temos dois número $Y=(69)_{10}$ e $Z=(12)_{10}$.

Suas representações em RNS seriam:

$$Y=(69 \text{ base}10) \text{ em RNS}(8 | 7 | 5 | 3) = (5 | 6 | 4 | 0)_{RNS(8|7|5|3)}$$

$$Z=(12 \text{ base}10) \text{ em RNS}(8 | 7 | 5 | 3) = (4 | 5 | 2 | 0)_{RNS(8|7|5|3)}$$

Para realizar multiplicações, basta multiplicarmos cada resídio referente à Y e Z em seus respectivos módulos.

$$\langle 5 \times 4 \rangle_8 = (20 \bmod 8) = 4$$

$$\langle 6 \times 5 \rangle_7 = (30 \bmod 7) = 2$$

$$\langle 4 \times 2 \rangle_5 = (8 \bmod 5) = 3$$

$$\langle 0 \times 0 \rangle_3 = 0$$

Já para a adição, , basta somarmos cada resídio referente à Y e Z em seus respectivos módulos.

$$\langle 5 + 4 \rangle_8 = (9 \bmod 8) = 1$$

$$\langle 6 + 5 \rangle_7 = (11 \bmod 7) = 4$$

$$\langle 4 + 2 \rangle_5 = (6 \bmod 5) = 1$$

$$\langle 0 + 0 \rangle_3 = 0$$

PROBLEMAS

Problema 4.6. Para umas entradas $Y=13_{10}$ e $Z=15_{10}$ faça as operações $(YxZ)_{RNS}$ e $(Y+Z)_{RNS}$ para os conjunto de módulos:

- a) $M1=\{3,5,7,17\};$
- b) $M2=\{16,15,17\};$
- c) $M3=\{7,13, 23\}.$

Problema 4.7. Para umas entradas $Y=16_{10}$ e $Z=9_{10}$ faça a operação $(YxZ+Y)_{RNS}$ para os conjunto de módulos:

- a) $M1=\{3,5,7,17\};$
- b) $M2=\{16,15,17\};$
- c) $M3=\{7,13, 23\}.$

SLIDE 22

Gabarito no Moodle

4 SISTEMAS DE NUMERAÇÃO RESIDUAL

- 4.1 Fundamentação RNS.
- 4.2 Codificação e Decodificação em RNS.
 - 4.2.1 Conversão Binário para RNS.
 - 4.2.2 Conversão RNS para Binário.
- 4.3 Aritmética em RNS.
- 4.4 Escolhendo o Módulo RNS**
- 4.5 Implementação em hardware.

Computer Arithmetic, Number Representation

SLIDE 23

Vejamos como realizar as escolhas dos módulos para aritmética em RNS.

4.4 ESCOLHENDO O CONJUNTO DE MÓDULOS

Faixa Dinâmica (DR) para nosso RNS: valores decimais [0, 100 000]

Estratégia 1: Para minimizar a escolha de módulos com valores grandes, e assim garantir aritmética de alta velocidade, escolha os números primos em sequência.

Escolher $m_0 = 2, m_1 = 3, m_2 = 5$, etc. até $m_5 = 13$:

| | | |
|-----------------------------------|---|---------------|
| RNS(13 11 7 5 3 2) | $M = 30030$ | Inadequado |
| RNS(17 13 11 7 5 3 2) | $M = 510510$ | Grande demais |
| RNS(17 13 11 7 3 2) | $M = 102102$ $5 + 4 + 4 + 3 + 2 + 1 = 19$ bits | Exato! |

Ajuste fino: combinar pares de módulos 2 & 13 (26) e 3 & 7 (21)

RNS(26 | 21 | 17 | 11) $M = 102102$

SLIDE 24

O conjunto de módulos escolhidos afeta tanto a eficiência quanto a complexidade do sistema. Em geral, tentamos fazer os módulos o menor possível, pois é a magnitude do maior módulo que dita a velocidade de operações aritméticas. Também frequentemente tentamos fazer com que todos os módulos sejam comparáveis em magnitude em relação ao maior, uma vez que, com a velocidade de cálculo já ditada pelos módulos maiores, geralmente não há vantagem em fragmentar o design através do uso de módulos muito pequenos.

Vamos considerar um exemplo em que desejamos realizar a representação de valores entre 0 e 100 000:

Uma estratégia simples é escolher os números primos em sequência até que a faixa dinâmica M se torne adequada. Assim, escolhemos $m_0 = 2, m_1 = 3, m_2 = 5, m_3 = 7, m_4 = 11$. Depois adicionamos $m_5 = 13$ à nossa lista, a faixa dinâmica se torna $M = 30030$.

Esta faixa dinâmica ainda está pequena. Então adicionamos $m_6 = 17$, obtendo faixa dinâmica de 510510.

Agora ela está muito grande. Podemos remover então o módulo 5 para obter a faixa igual a 102102.

Para um ajuste fino, podemos combinar os módulos (13 e 2) e (7 e 3) para obter a solução de módulos RNS(26 | 21 | 17 | 11).

4.4 ESCOLHENDO O CONJUNTO DE MÓDULOS

Intervalo de destino para nosso RNS: valores decimais [0, 100 000]

Estratégia 2: Para simplificar as operações modulares ($\text{mod } m_i$), escolher unicamente módulos da forma 2^a ou $2^a - 1$, “Módulos de baixo custo”

$$\text{RNS}(2^{a_{k-1}} | 2^{a_{k-2}} - 1 | \dots | 2^{a_1} - 1 | 2^{a_0} - 1)$$

Podemos ter unicamente módulos ímpar
 $2^{a_i} - 1$ e $2^{a_j} - 1$ são coprimos se a_i e a_j são primos

| | | |
|--|----------------|--------------|
| RNS($2^3 2^3 - 1 2^2 - 1$) | bases: 3, 2 | $M = 168$ |
| RNS($2^4 2^4 - 1 2^3 - 1$) | bases: 4, 3 | $M = 1680$ |
| RNS($2^5 2^5 - 1 2^3 - 1 2^2 - 1$) | bases: 5, 3, 2 | $M = 20832$ |
| RNS($2^5 2^5 - 1 2^4 - 1 2^3 - 1$) | bases: 5, 4, 3 | $M = 104160$ |

Comparação

| | | |
|--|---------|--------------|
| RNS($15 13 11 2^3 7$) | 18 bits | $M = 120120$ |
| RNS($2^5 2^5 - 1 2^4 - 1 2^3 - 1$) | 17 bits | $M = 104160$ |

SLIDE 25

Observe agora a segunda estratégia para escolha de módulos, no slide acima. De forma análoga vamos realizando tentativas até chegar próximo do valor de 100 000 como especificado no problema.

Começamos com a primeira tentativa com os módulos RNS($2^3 = 8 | 2^3 - 1 = 7 | 2^2 - 1 = 3$) atingindo uma faixa dinâmica de 168 e finalizamos com os módulos RNS($2^5 = 32 | 2^5 - 1 = 31 | 2^4 - 1 = 15 | 2^3 - 1 = 7$), atingindo um valor de 104160 para faixa dinâmica.

4.4 ESCOLHENDO O CONJUNTO DE MÓDULOS

Intervalo de destino para nosso RNS: valores decimais [0, 100 000]

Estratégia 3: Para simplificar as operações modulares ($\text{mod } m_i$), escolher módulos das forma 2^a , $2^a - 1$, ou $2^a + 1$

$$\text{RNS}(2^{a_{k-1}} | 2^{a_{k-2}} \pm 1 | \dots | 2^{a_1} \pm 1 | 2^{a_0} \pm 1)$$

Podemos ter unicamente módulos ímpar
 $2^{a_i} - 1$ e $2^{a_j} + 1$ são primos

$$\begin{array}{ll} \text{RNS}(2^5 | 2^4-1 | 2^4+1 | 2^3-1) & M = 57120 \\ \text{RNS}(2^5 | 2^4+1 | 2^3+1 | 2^3-1 | 2^2-1) & M = 102816 \end{array}$$

O Modulo $2^a + 1$ não é tão conveniente como $2^a - 1$
(precisa de um bit mais de resíduo e as operações modulares não são tão simples)

Apr. 2012
Computer Arithmetic, Number Representation

SLIDE 26

Como terceita estratégia utilizaremos módulos na forma 2^a , $2^a - 1$, ou $2^a + 1$

Observe a Solução dada, que de forma similar utiliza a tentativas para aproximar a faixa dinâmica.

4.4 ESCOLHENDO O CONJUNTO DE MÓDULOS

Foram mostradas 3 estratégias para seleção de modulo tendo em consideração os módulos mais eficientes.

A solução parece ir no uso de conjunto com muitos módulos para minimizar o número de bits por canal modular. No entanto quanto mais módulos no conjunto a conversão final RNS-bin será mais complexa.

Aqui é mostrado a escolha de conjunto de módulos para conversores RNS-bin eficientes:

| Conjunto de módulos | DR | Ano |
|---|-------------|------|
| $\{2^n, 2^n \pm 1\}$ | $3n$ | 2002 |
| $\{2^{2n}, 2^n \pm 1\}$ | $4n$ | 2004 |
| $\{2^n \pm 1, 2^n \pm 3\}$ | $4n$ | 2004 |
| $\{2^n, 2^n \pm 1, 2^n \pm 2^{(n+1)/2} + 1\}$ | $5n$ | 2005 |
| $\{2^n, 2^n \pm 1, 2^{n+1} - 1\}$ | $5n$ | 2007 |
| $\{2^{2n}, 2^n \pm 1, 2^{2n} + 1\}$ | $6n$ | 2010 |
| $\{2^n, 2^n \pm 1, 2^n \pm 2^{(n+1)/2} + 1, 2^{n+1} + 1\}$ | $6n + 1$ | 2013 |
| $\{2^{2n}, 2^n \pm 1, 2^n \pm 2^{(n+1)/2} + 1, 2^{n+1} + 1\}$ | $7n + 1$ | 2013 |
| $\{2^{3n}, 2^n \pm 1, 2^n \pm 2^{(n+1)/2} + 1, 2^{n+1} + 1\}$ | $8n + 1$ | 2013 |
| $\{2^{2n}, 2^n \pm 1, 2^n \pm k_1, 2^n \pm k_2, \dots, 2^n \pm k_f\}$ | $(2f + 3)n$ | 2014 |

SLIDE 27

Vejamos os conjuntos de módulos já estudados e sua faixa de dinâmica (DR).

4.4 ESCOLHENDO O CONJUNTO DE MÓDULOS

Foram mostradas 3 estratégias para seleção de modulo tendo em consideração os módulos mais eficientes.

A solução parece ir no uso de conjunto com muitos módulos para minimizar o número de bits por canal modular. No entanto quanto mais módulos no conjunto a conversão final RNS-bin será mais complexa.

Aqui é mostrado a escolha de conjunto de módulos para conversores RNS-bin eficientes:

| | Conjunto de módulos | DR | Ano |
|----|---|-------------|------|
| M1 | $\{2^n, 2^n \pm 1\}$ | $3n$ | 2002 |
| | $\{2^{2n}, 2^n \pm 1\}$ | $4n$ | 2004 |
| M2 | $\{2^n \pm 1, 2^n \pm 3\}$ | $4n$ | 2004 |
| | $\{2^n, 2^n \pm 1, 2^n \pm 2^{(n+1)/2} + 1\}$ | $5n$ | 2005 |
| | $\{2^n, 2^n \pm 1, 2^{n+1} - 1\}$ | $5n$ | 2007 |
| | $\{2^{2n}, 2^n \pm 1, 2^{2n} + 1\}$ | $6n$ | 2010 |
| | $\{2^n, 2^n \pm 1, 2^n \pm 2^{(n+1)/2} + 1, 2^{n+1} + 1\}$ | $6n + 1$ | 2013 |
| | $\{2^{2n}, 2^n \pm 1, 2^n \pm 2^{(n+1)/2} + 1, 2^{n+1} + 1\}$ | $7n + 1$ | 2013 |
| | $\{2^{3n}, 2^n \pm 1, 2^n \pm 2^{(n+1)/2} + 1, 2^{n+1} + 1\}$ | $8n + 1$ | 2013 |
| | $\{2^{2n}, 2^n \pm 1, 2^n \pm k_1, 2^n \pm k_2, \dots, 2^n \pm k_f\}$ | $(2f + 3)n$ | 2014 |

Vamos usar os conjuntos como estudos de caso

SLIDE 28

Aqui estão alguns exemplos de baixo custo.

PROBLEMAS

Problema 4.8. Aplique as três estratégias apresentadas na teoria para obter uma Faixa Dinâmica (DR) com valores de saída [0, 200 000].

Problema 4.12. Obtenha conjuntos modulares válidos com faixa dinâmica de 20 bits DR=[0, 1048 576] e com $n=5$ bits por canal (máximo). Indique a faixa dinâmica das estruturas RNS e compare com a eficiência da representação com binário.

SLIDE 29

Gabarito no Moodle

4 SISTEMAS DE NUMERAÇÃO RESIDUAL

- 4.1 Fundamentação RNS.**
- 4.2 Codificação e Decodificação em RNS.**
 - 4.2.1 Conversão Binário para RNS.**
 - 4.2.2 Conversão RNS para Binário.**
- 4.3 Aritmética em RNS.**
- 4.4 Escolhendo o Módulo RNS.**
- 4.5 Implementação em hardware.**

Computer Arithmetic, Number Representation

SLIDE 30

Por fim, encontraremos as soluções de implementação em hardware do assunto que foi visto.

4.5 IMPLEMENTAÇÃO EM HARDWARE

$M1 = \{2^{2n}, 2^n - 1, 2^n + 1\}$

1. Conversor Binário-RNS (Direto) ←
2. Unidade aritmética RNS (somadores e multiplicadores) ← A ser visto em outros capítulos
3. Conversor RNS-Binário (Reverso) ←

$M2 = \{2^{2n}, 2^n - 3, 2^n + 3\}$

1. Conversor Binário-RNS (Direto) ←
2. Unidade aritmética RNS (somadores e multiplicadores) ← A ser visto em outros capítulos
3. Conversor RNS-Binário (Reverso) ←

SLIDE 31

Implementações eficientes em Hardware podem ser alcançadas por meio da utilização de alguns conjuntos de módulos. Aqui usaremos:

$M1 = \{2^{2n}, 2^n - 1, 2^n + 1\}$

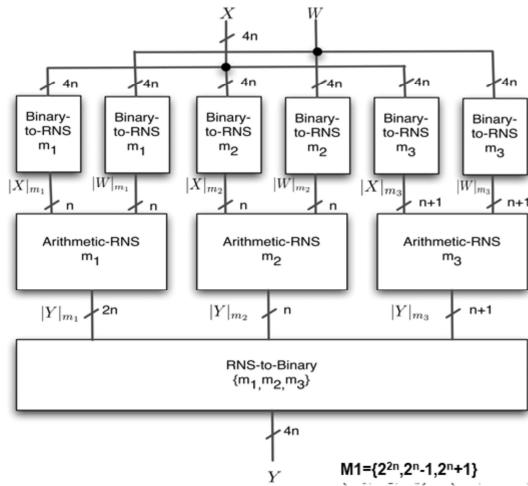
$M2 = \{2^{2n}, 2^n - 3, 2^n + 3\}$

Exemplos já apontados com sendo bons conjuntos (slide 28).

M1={2²ⁿ, 2ⁿ⁻¹, 2ⁿ⁺¹}: HARDWARE IMPLEMENTATION

- Blocos básicos

1. Conversor direto
2. Unidade aritmética
3. Conversor reverso



SLIDE 32

Considere que faremos alguma operação aritmética entre dois número X e W.

Para isso teremos que realizar a conversão direta de binário para RNS. A conversão direta será realizada em função dos módulos escolhidos. Em seguida faremos a operação aritmética desejada e por fim converteremos o resultado RNS de volta para binário. Observe que este conjunto de módulo vai requerer tamanhos diferentes de bits para as operações em cada módulo.

O módulo 2^{2n} vai requerer uma saída de tamanho $2n$ após a computação da operação aritmética, enquanto que $2^{n-1}, 2^{n+1}$ irão requerer apenas um tamanho (n) e ($n+1$) respectivamente.

M1={2²ⁿ, 2ⁿ-1, 2ⁿ+1}: HARDWARE IMPLEMENTATION CONVERSÃO DIRETA

Um numero inteiro $X = \{x_{(4n-1)}, \dots, x_1, x_0\}$ pode ser expressado em notação binaria como:

$$X = \sum_{i=1}^{4n-1} 2^i x_i = 2^{3n} N_3 + 2^{2n} N_2 + 2^n N_1 + N_0, \quad (1)$$

onde os arrays $N_3 = \{x_{(4n-1)}, \dots, x_{(3n+1)}, x_{3n}\}$, $N_2 = \{x_{(3n-1)}, \dots, x_{(2n+1)}, x_{2n}\}$, $N_1 = \{x_{(2n-1)}, \dots, x_{(n+1)}, x_n\}$ e $N_0 = \{x_{(n-1)}, \dots, x_1, x_0\}$. Usando notação binaria e conjunto de módulos $\{m_1, m_2, m_3\} = \{2^{2n}, 2^n - 1, 2^n + 1\}$, a faixa dinâmica do valor X é $[0, M - 1]$, onde $M = m_1 m_2 m_3$. Três conversores são necessários de modo a obter a representação do RNS, um para cada elemento de base.

SLIDE 33

Então vejamos o que diz o slide. Observemos que será preciso 3 conversores (1 para cada módulo), para conversão direta.

$M1=\{2^{2n}, 2^n-1, 2^n+1\}$: HARDWARE IMPLEMENTATION CONVERSÃO DIRETA

- Canal $m_1 = 2^{2n}$: O canal mais simples é o conversor usando o módulo m_1 . O valor $|X|_{m_1}$ pode ser obtido pelo resto da divisão de X por 2^{2n} , o que pode ser conseguido por meio de truncar o valor de X , uma vez que:

$$|X|_{m_1} = \overbrace{|2^{3n}|_{m_1}}^{=0} N_3 + \overbrace{|2^{2n}|_{m_1}}^{=0} N_2 + 2^n N_1 + N_0 = \{x_{(2n-1)}, \dots, x_1, x_0\}. \quad (2)$$

- Canal $m_2 = 2^n - 1$: Devido a que $|2^n|_{2^n-1} = 1$, podemos expressar a Eq. 1 como:

$$|X|_{m_2} = |N_3 + N_2 + N_1 + N_0|_{2^n-1} = |N_3 + |N_2 + N_1 + N_0|_{2^n-1}|_{2^n-1}. \quad (3)$$

- Canal $m_3 = 2^n + 1$: Devido a que $|2^n|_{2^n+1} = -1$, podemos expressar a Eq. 1 como:

$$|X|_{m_3} = |N_3 - N_2 + N_1 - N_0|_{2^n+1} = |-N_3 + |N_2 - N_1 + N_0|_{2^n+1}|_{2^n+1}. \quad (4)$$

SLIDE 34

Vejamos um exemplo utilizando o módulo 4.

Os pesos de cada bit serão:

$$2^3, 2^2, 2^1, 2^0$$

$$\begin{array}{cccc} 8 & 4 & 2 & 1 \end{array}$$

Aplicando o módulo no pesos:

$$\begin{array}{cccc} 0 & 0 & 2 & 1 \end{array}$$

Então podemos truncar o valor nos 2 primeiros bits pois os demais serão zero:

Já para um módulo igual a 2^n-1 , por exemplo módulo 7 com $n=3$

$$2^3, 2^2, 2^1, 2^0$$

$$\begin{array}{cccc} 8 & 4 & 2 & 1 \end{array}$$

Aplicando o módulo no pesos:

$$\begin{array}{cccc} 1 & 4 & 2 & 1 \end{array}$$

O valor do peso 2^3 aplicado ao módulo 7 é igual a 1. Então podemos reinserir esse valor somando na posição 2^0

De forma similar 2^n+1 terá o valor -1 reinserido na matriz de cálculo.

M1={ 2^{2n} , 2^n-1 , 2^n+1 }: HARDWARE IMPLEMENTATION CONVERSÃO DIRETA

- **Bloco 2^{2n}** : Truncamento a partir do dígito 2^{2n} , pois $(2^{2n} \bmod 2^{2n}) = 0$
- **Bloco $2^n - 1$** : Soma dos N termos, posicionando o carry em 2^0 (EAC). $(2^n \bmod 2^n - 1) = 1$
- **Bloco $2^n + 1$** : Somatório dos N termos, posicionando o complemento do carry em 2^0 (IEAC). $(2^n \bmod 2^n + 1) = -1$

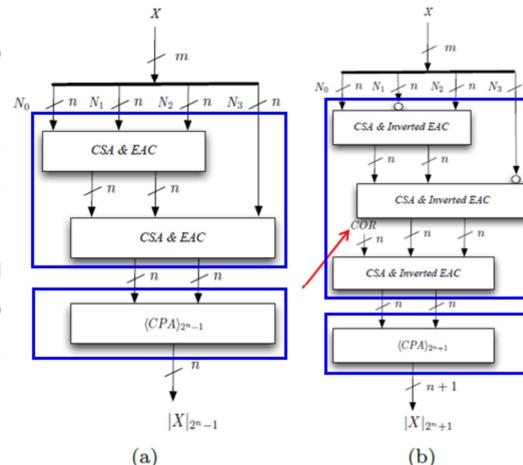


Figura 2 – Conversores diretos $2n - 1$

SLIDE 35

Para entender melhor os circuitos formados pode-se encontrar um exemplo de como são truncados, ou reposicionados os carry no Moodle.

Caminho no Moodle:

Capítulo 4 -> Gabarito_problemas_capítulo_4 -> Problema_4_10 ->
Compressors.pdf e também o arquivo Adders.pdf

M1={ 2^{2n} , 2^n-1 , 2^n+1 }: HARDWARE IMPLEMENTATION CONVERSÃO DIRETA

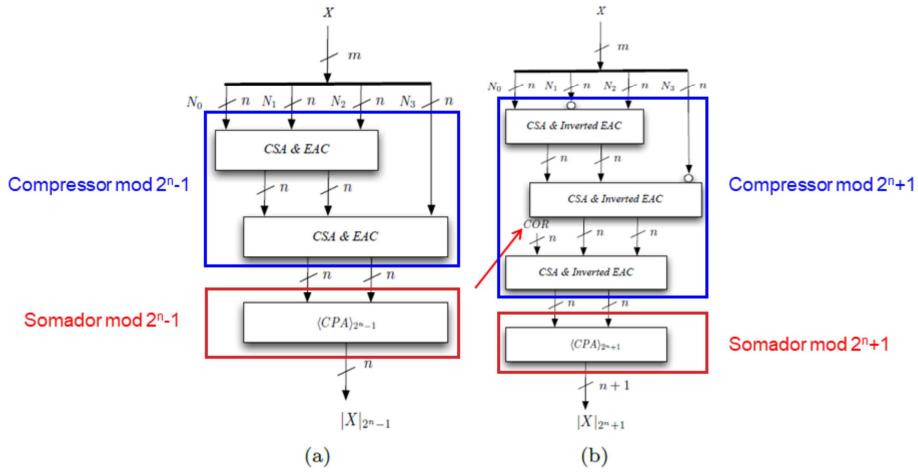


Figura 2 – Conversores diretos $2^n - 1$

SLIDE 36

Essas arquiteturas são formadas de uma etapa de compressão, que somará todos os bits e resultará em 2 vetores de bits finais e em seguida faremos a soma final desses dois vetores restantes.

M1={2^{2N}, 2^{N-1}, 2^{N+1}} : HARDWARE IMPLEMENTATION CONVERSÃO REVERSA

Para cada conjunto de módulos implementado, é necessário que seja projetado um conversor reverso.

Aplicando o algoritmo Novo CRT-I:

$$X = \left| \sum_{i=1}^n |V_i R_i|_{\hat{m}_1} \right|_{\hat{m}_1} m_1 + R_1 \quad \text{, onde} \quad V_1 = \frac{|\hat{m}_1^{-1}|_{m_1} \hat{m}_1 - 1}{m_1} \\ V_i = |\hat{m}_i^{-1}|_{m_i} \frac{\hat{m}_i}{m_i} \quad \text{for} \quad 2 \leq i \leq n$$

Solução obtida no Problema 4.5 para n=4

SLIDE 37

Já para conversão reversa resolveremos o algoritmo como comentado em alguns slides anteriores.

Na prática formaremos um circuito que realizará um conjunto de multiplicações por constante e depois somará esses resultados.

Lembrando o exemplo do slide 17:

$\hat{m}_r = (255, 272, 240)$

Multiplicativa inversa: (15,8,9)

Então o circuito realizará: A soma das multiplicações $[(255 \times 15 \times \text{Resíduo_1}) + (272 \times 8 \times \text{Resíduo_2}) + (240 \times 9 \times \text{Resíduo_3})]$ aplicado ao módulo M, para chegar no resultado.

PROBLEMAS

Problema 4.10. Considere o seguinte conjunto de módulos $\{2^{2n}, 2^n-1, 2^n+1\}$, para $n=4$ e uma entrada-saída de $4n$ bits:

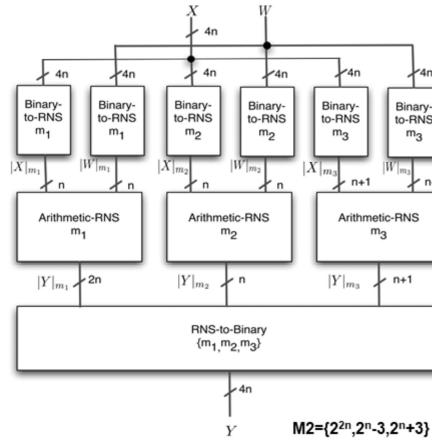
- a) Obtenha a estrutura para fazer a conversão binário-RNS (use compressores e somadores modulo 15 e 17).
- b) Obtenha a estrutura para fazer a conversão RNS-binário (use o algoritmo novo CRT-I, compressores e somadores módulo 255).
- c) Indique a faixa dinâmica da estrutura RNS e compare com a eficiência da representação com binário.

SLIDE 38

Gabarito no Moodle.

M2={2²ⁿ , 2ⁿ⁻³, 2ⁿ⁺³}: HARDWARE IMPLEMENTATION

- Blocos básicos
 1. Conversor direto
 2. Unidade aritmética
 3. Conversor reverso



SLIDE 39

Existem ainda outros conjuntos de módulos. Entretanto as arquiteturas seguem a mesma lógica.

Conversor direto
Unidade Aritmética
Conversor Reverso

M2: HARDWARE IMPLEMENTATION CONVERSÃO DIRETA

Para conjunto :

- **Bloco 2^{2n} :** Truncamento a partir do digito 2^{2n} ,
pois $(2^{2n} \bmod 2^{2n}) = 0$
- **Bloco $2^n - k$:** Soma dos N termos, cada *carry*
tem um peso k. $(2^n \bmod 2^n - 1) = k$
- **Bloco $2^n + k$:** Somatório dos N termos, cada
carry tem um peso -k $(2^n \bmod 2^n + 1) = -k$

SLIDE 40

A diferença então é que para módulos $2^n - k$ teremos carries que serão reinseridos nas matrizes de cálculos iguais a k ou -k

M2: HARDWARE IMPLEMENTATION CONVERSOR RNS-BIN

A representação RNS pode ser convertida de volta para binário (X) usando:

a) Teorema Chinês do Resto (CRT):

$$X = \left| \sum_{i=1}^n \hat{m}_i \left| \hat{m}_i^{-1} \right|_{m_i} \times R_i \right|_M , \text{ onde } M = \prod_{i=1}^n m_i$$

$\hat{m}_i^{-1} \mid_{m_i}$ represents the multiplicative inverse of \hat{m}_i with respect to modulus m_i

$\hat{m}_i = M/m_i$

b) Novo CRT-I:

$$X = \left| \sum_{i=1}^n |V_i R_i|_{\hat{m}_1} \right|_{\hat{m}_1} m_1 + R_1 , \text{ onde } V_1 = \frac{|\hat{m}_1^{-1}|_{m_1} \hat{m}_1 - 1}{m_1}$$

$$V_i = |\hat{m}_i^{-1}|_{m_i} \frac{\hat{m}_i}{m_1} \quad \text{for} \quad 2 \leq i \leq n$$

SLIDE 41

E de forma similar utilizaremos os mesmos algoritmos para conversão reversa.

PROBLEMAS

Problema 4.11. Considere o seguinte conjunto de módulos $\{2^n, 2^{n-3}, 2^{n+3}\}$, para $n=4$ e uma entrada-saída de $3n$ bit:

- a) Obtenha a estrutura para fazer a conversão binário-RNS (use compressores e somadores modulo 13 e 19).
- b) Obtenha a estrutura para fazer a conversão RNS-binário binário (use novo CRT-I, compressores e somadores módulo 247).
- c) Indique a faixa dinâmica da estrutura RNS e compare com a eficiência da representação com binário.

SLIDE 42

Gabarito no Moodle.