RESIDUE NUMBER SYSTEM

(introduction to hardware aspects)

Dr. Danila Gorodecky

danila.gorodecky@gmail.com

Terminology

– **Residue number system (RNS) (refers to *Chinese remainder theorem*)**

– **Residue numeral system (RNS)**

– **Modular arithmetic (MA) (refers to *moduli – X (mod P)*)**

– **Complete residue system**

– **Clock arithmetic (refers to 12-hour arrow clock in which numbers "wrap around“ upon reaching the modulo)**

Milestones

– Chinese mathematician Sunzi Suanjing proposed a theorem (Chinese remainder theorem) in the 3rd century AD;

– the theorem was generalized by Chinese mathematician Qin Jiushao in 1247;

– first real implementation of the theorem by German mathematician Carl Gauss in 1801 "to find the years that have a certain period number with respect to the solar and lunar cycle and the Roman indiction“;

– first implementation in computer science by Czechoslovakian engineer Miro Valach in 1955 “Origin of the code and number system of remainder classes”, *Stroje Na ZpracovaniInformaci,* vol. 3, Nakl. CSAV, Prague.

– Processing of results of the Unified State Exam (utilized to entrance to University in Russia;

– Digital filternig with finite impulse response (FIR-filtering);

– Crypto system of Federal Reserve System of USA;

– Air Defense System (USA, Russia);

– cryptography in Space (Russia);

– Space flight control (Russia)

Let's $p_1, p_2, \ldots, p_n$ are positive integers (are often called as moduli) such, that greatest common divisor for a couple (p_i, p_j) equals '1'.

Then the system

$$
\begin{aligned}
y &= x_1 \pmod{p_1} \\
y &= x_2 \pmod{p_2} \\
&\ldots \\
y &= x_n \pmod{p_n}
\end{aligned}
$$

has a simultaneous solution which is unique modulo

$$p_1, p_2, \ldots, p_n$$

Example for the Theorem

$$P = p_1 \cdot p_2 \cdot p_3 = 5 \cdot 7 \cdot 9 = 315$$

We can express an arbitrary number definitely in the scope from 0 to 314

Let's $A = 100$ **, hence**

$$A = 100 = 0 \pmod{5}$$
$$A = 100 = 2 \pmod{7}$$
$$A = 100 = 1 \pmod{9}$$

and $A = (0,2,1)$ **in the RNS representation**

RNS is not *positional numeral system*

$$\left(100(\mathrm{mod}\ 5),\ 100(\mathrm{mod}\ 7),\ 100(\mathrm{mod}\ 9)\right) = (0,\ 2,\ 1) = 100$$

$$\left(100(\mathrm{mod}\ 7),\ 100(\mathrm{mod}\ 9),\ 100(\mathrm{mod}\ 5)\right) = (2,\ 1,\ 0) = 100$$

Binary system

$$(1100100)_2 = 100_{10}$$

$$(0110100)_2 = 52_{10}$$

Example of the computation in RNS

$$P = p_1 \cdot p_2 \cdot p_3 = 5 \cdot 7 \cdot 9 = 315$$

$$A + B = 100 + 13 = S$$

1) $A = (0,2,1) \quad B = (3,6,4)$

2) $A + B =$

$$= ((0+3)\ (\mathrm{mod}\ 5),\ (2+6)\ (\mathrm{mod}\ 7),\ (1+4)\ (\mathrm{mod}\ 9)) =$$

$$= (3\ (\mathrm{mod}\ 5),\ 1\ (\mathrm{mod}\ 7),\ 5\ (\mathrm{mod}\ 9)) =$$

$$= (3,1,5)$$

Example of the computation in RNS

3) $S = S_1 \cdot Y_1 + S_2 \cdot Y_2 + S_3 \cdot Y_3 - r \cdot P$

$Y_i = \left(\frac{P}{p_i}\right) k_i; \quad \frac{Y_i}{p_i} = 1 \pmod{p_i}; \quad r \cdot P \le S_1 \cdot Y_1 + S_2 \cdot Y_2 + S_3 \cdot Y_3 < (r+1) \cdot P$

a) $Y_1 = \left(\frac{315}{5}\right) \cdot k_1 = 63 \cdot k_1$ and $\frac{63 \cdot k_1}{5} = 1 \pmod{5}$, then $k_1 = 2$ and $Y_1 = 126$

b) $Y_2 = \left(\frac{315}{7}\right) \cdot k_2 = 45 \cdot k_2$ and $\frac{45 \cdot k_2}{7} = 1 \pmod{7}$, then $k_2 = 5$ and $Y_2 = 225$

c) $Y_3 = \left(\frac{315}{9}\right) \cdot k_3 = 35 \cdot k_3$ and $\frac{35 \cdot k_3}{9} = 1 \pmod{9}$, then $k_3 = 8$ and $Y_3 = 280$

d) $r \cdot 315 \le 3 \cdot 126 + 8 \cdot 225 + 5 \cdot 280 < (r+1) \cdot 315$, then $r = 6$

$$S = 3 \cdot 126 + 8 \cdot 225 + 5 \cdot 280 - 6 \cdot 315 =$$

$$= 2003 - 1890 = 113$$

1) What is maximum bit range of *A* and *B* should be chosen for unambiguous representation *A* + *B* = *S* in RNS with moduli 11,13, and 15?

$P = 11 * 13 * 15 = 2145$

$0 \leq S < 2145$ and *S* is *12*-bit number.

Hence, in order to represent *A* + *B* = *S*, *A* and *B* should be limited 11-bit tuples, when *A* and *B* both equal 1077.

2) What is maximum bit range of *A, B,* and *C* can be used for unambiguous representation *A* * *B* * *C* = *R* in RNS with moduli 11,13, and 15?

$P = 11 * 13 * 15 = 2145$

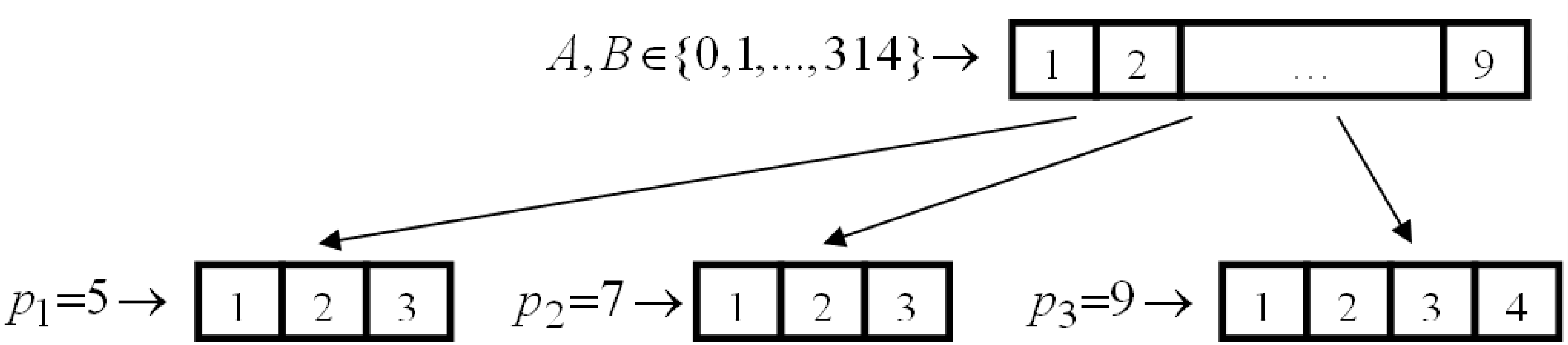
$0 \leq S < 2145$ and *S* is *12*-bit number.

Hence, in order to represent A * B * C = R, *A, B,* and *C* should be limited 4-bit tuples, when A, B, and C equal 12.

Main advantage of RNS

Significantly smaller ranges of numbers in arithmetic calculations than initial numbers

Example of the computation in RNS



$$A = \{A_1, A_2, A_3\}$$

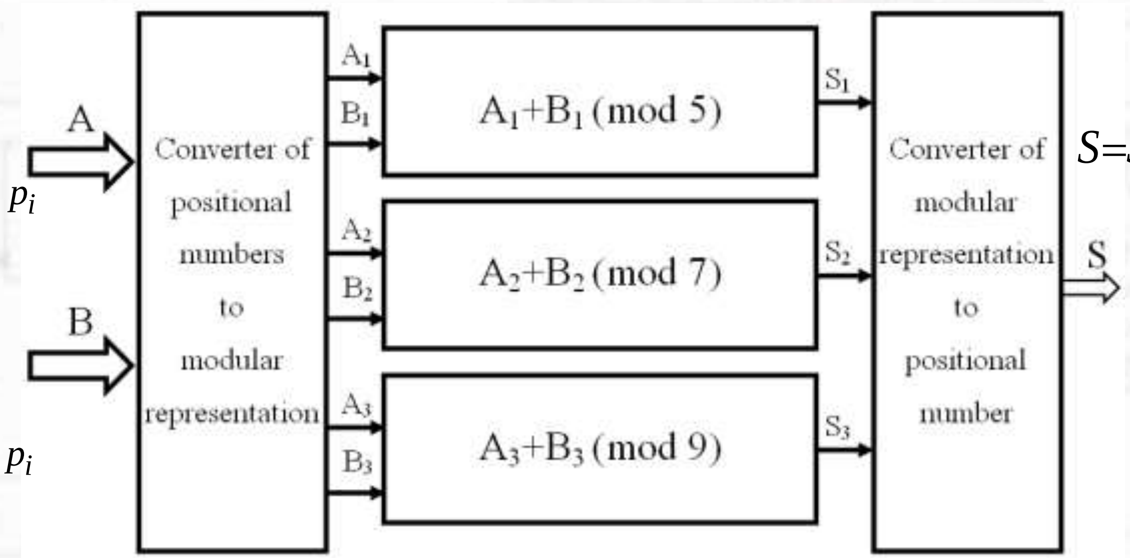
$$A_i \pmod{p_i} = A - \left[\frac{A}{p_i}\right] p_i$$

$$0 \le A \le 314$$

$$B = \{B_1, B_2, B_3\}$$

$$B_i \pmod{p_i} = B - \left[\frac{B}{p_i}\right] p_i$$

$$0 \le B \le 314$$



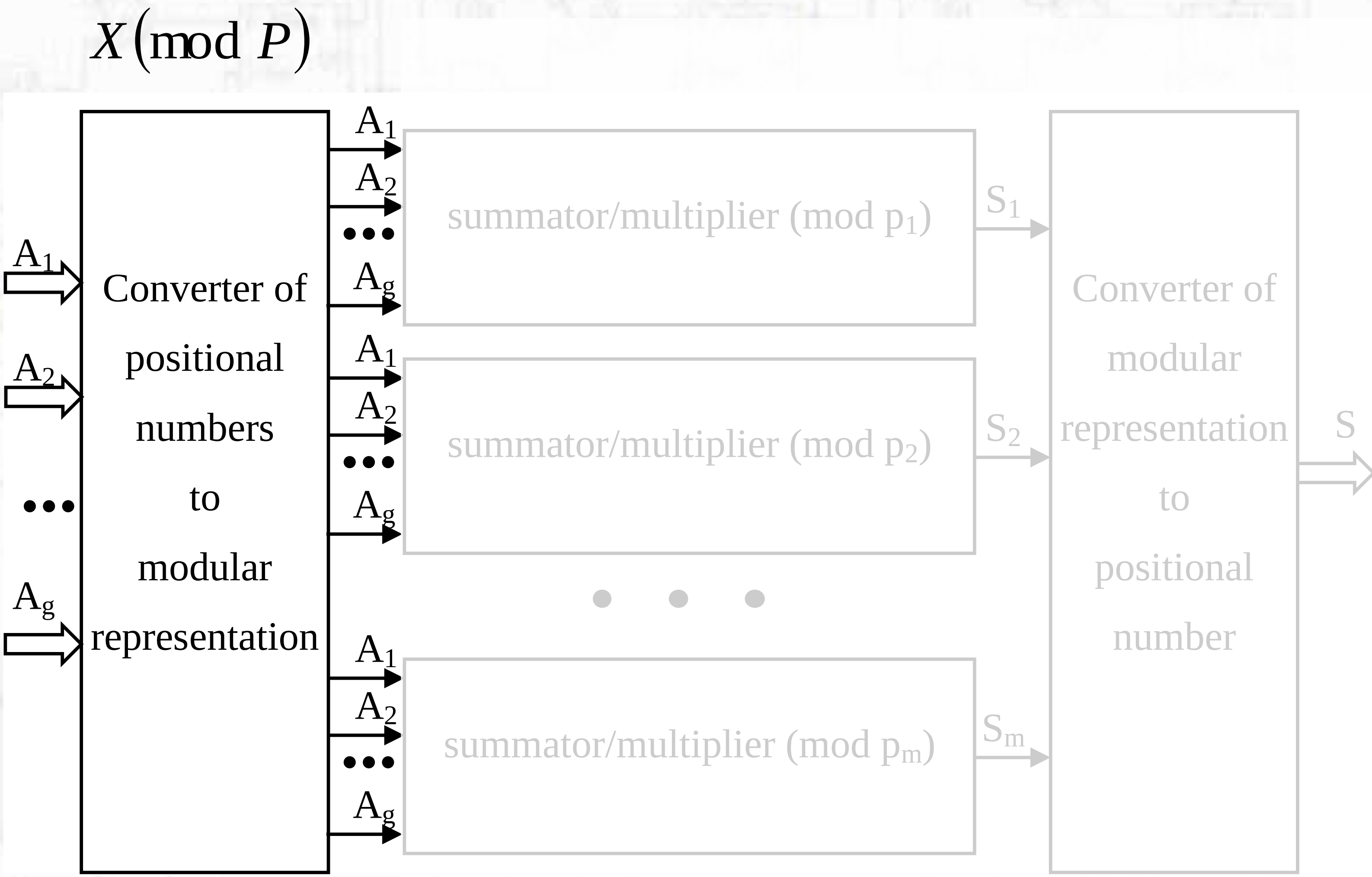
$$S = S_1 \cdot Y_1 + S_2 \cdot Y_2 + S_3 \cdot Y_3 - r \cdot P$$

$$\frac{Y_i}{p_i} = 1 \pmod{p_i}$$

$$Y_i = \left(\frac{P}{p_i}\right) k_i$$

$i = 1, 2, 3 \qquad k_i = \overline{1, p_i} \qquad r = 0, 1, 2, \ldots$

X (mod P) hardware calculations in RNS



Pipelining approach of X (mod P) calculations

1) Pipelining (iterative) approach. It is based on the formula:

$$X = P \cdot Q + A = P \cdot 2^{\delta} \cdot q_{\delta} + P \cdot 2^{\delta-1} \cdot q_{\delta-1} + \ldots + P \cdot 2^{0} \cdot q_{0} + A$$

$$X = \left(x_{\varphi}, x_{\varphi-1}, \ldots, x_{1}\right) \qquad P = \left(p_{\gamma}, p_{\gamma-1}, \ldots, p_{1}\right)$$

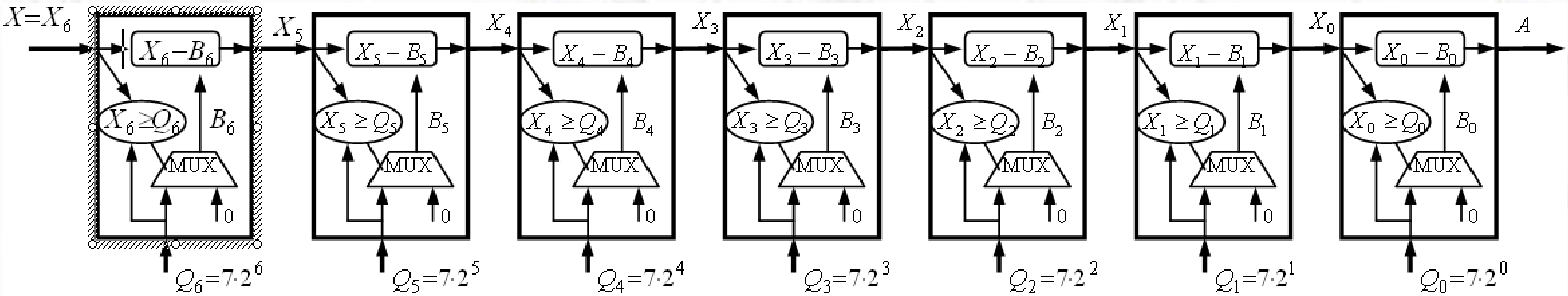
$$P \cdot 2^{\delta+1} > 2^{\varphi} - 1 \geq P \cdot 2^{\delta}$$

Pipelining approach of X (mod P) calculations

$$510 \pmod 7 = (7 \cdot Q + A) \pmod 7 =$$
$$= (7 \cdot 2^6 \cdot q_6 + 7 \cdot 2^5 \cdot q_5 + 7 \cdot 2^4 \cdot q_4 + 7 \cdot 2^3 \cdot q_3 + 7 \cdot 2^2 \cdot q_2 + 7 \cdot 2^1 \cdot q_1 + 7 \cdot 2^0 \cdot q_0 + A) \pmod 7 =$$
$$= A \pmod 7$$

1) $X_6 \geq Q_5$, i.e. $510 \geq 448$, then $B_6 = Q_6 = 448$ and $X_6 - B_6 = 510 - 448 = 62 = X_5$;
2) $X_5 < Q_5$, i.e. $62 < 224$, then $B_5 = Q_5 = 0$ and $X_5 - B_5 = 62 - 0 = 62 = X_4$;
3) $X_4 < Q_4$, i.e. $62 < 112$, then $B_4 = Q_4 = 0$ and $X_4 - B_4 = 62 - 0 = 62 = X_3$;
4) $X_3 \geq Q_3$, i.e. $62 \geq 56$, then $B_3 = Q_3 = 56$ and $X_3 - B_3 = 62 - 56 = 6 = X_2$;
5) $X_2 < Q_2$, i.e. $6 < 28$, then $B_2 = Q_2 = 0$ and $X_2 - B_2 = 6 - 0 = 6 = X_1$;
6) $X_1 < Q_1$, i.e. $6 < 14$, then $B_1 = Q_1 = 0$ and $X_1 - B_1 = 6 - 0 = 6 = X_0$;
7) $X_0 < Q_0$, i.e. $6 < 7$, then $B_0 = Q_0 = 0$ and $X_0 - B_0 = 6 - 0 = 6 = A$

$$510 \pmod 7 = 7 \cdot Q + A = (7 \cdot 2^6 + 7 \cdot 2^3 + A) \pmod 7 = 6 \pmod 7$$



2) Recursive approach. It is suitable for special moduli, e.c. $2^n \pm 1$ and $2^n \pm 3$

$$X = x_1 + 2x_2 + 2^2 x_3 + \ldots + 2^{\gamma-1} x_\gamma + 2^\gamma x_{\gamma+1} + \ldots \qquad x_1, x_2, \ldots, x_\gamma, \ldots \in \{0, 1\}$$

$$(x_1, x_2, \ldots, x_\gamma) = X_1 \quad (x_{\gamma+1}, x_{\gamma+2}, \ldots, x_{2\gamma}) = X_2 \quad \ldots$$

$$X = X_1 + 2^\gamma X_2 + 2^{2\gamma} X_3 + \ldots$$

as $2^\gamma = 1 \left(\bmod\ 2^\gamma - 1\right)$**, so** $X \left(\bmod\ 2^\gamma - 1\right) = (X_1 + X_1 + X_3 + \ldots)\left(\bmod\ 2^\gamma - 1\right)$

Example. ***X (mod P)***, where $X = 437 = (110110101)$ and $P = 2^3 - 1 = 7$

a) $(110110101) = ((110) + (110) + (101))(\bmod 7) = (10001)(\bmod 7) =$

b) $= ((010) + (001))\ (\bmod\ 7) = (011)\ (\bmod\ 7) = 3\ (\bmod\ 7)$

3) It is suitable for an arbitrary modulo and is based on the next formula:

$$X = \left(2^0 \cdot x_1 + 2^1 \cdot x_2 + 2^2 \cdot x_3 + 2^3 \cdot x_4 + \ldots\right)(\text{mod}\, P) =$$
$$= 2^0 \cdot x_1 (\text{mod}\, P) + 2^1 \cdot x_2 (\text{mod}\, P) + 2^2 \cdot x_3 (\text{mod}\, P) + 2^3 \cdot x_4 (\text{mod}\, P) + \ldots$$

$$x_1, x_2, x_3, x_4, \ldots \in \{0,1\}$$

Example. ***X (mod P)***, where $X = (x_1, x_2, \ldots, x_{10})$ and $P = 23$

$$X = \left(2^0 \cdot x_1 + 2^1 \cdot x_2 + 2^2 \cdot x_3 + 2^3 \cdot x_4 + 2^4 \cdot x_5 + 2^5 \cdot x_6 + 2^6 \cdot x_7 + 2^7 \cdot x_8 + 2^8 \cdot x_9 + 2^9 \cdot x_{10}\right)(\text{mod}\, 23) =$$
$$= x_1 (\text{mod}\, 23) + 2 \cdot x_2 (\text{mod}\, 23) + 4 \cdot x_3 (\text{mod}\, 23) + 8 \cdot x_4 (\text{mod}\, 23) + 16 \cdot x_5 (\text{mod}\, 23) +$$
$$+ 9 \cdot x_6 (\text{mod}\, 23) + 18 \cdot x_7 (\text{mod}\, 23) + 13 \cdot x_8 (\text{mod}\, 23) + 3 \cdot x_9 (\text{mod}\, 23) + 6 \cdot x_{10} (\text{mod}\, 23)$$

- If $0 \le S < 23 \Rightarrow X(\text{mod}\, 23) = X$
- If $23 \le S < 46 \Rightarrow X(\text{mod}\, 23) = X - P$
- If $46 \le S < 69 \Rightarrow X(\text{mod}\, 23) = X - 2 \cdot P$
- If $69 \le S < 92 \Rightarrow X(\text{mod}\, 23) = X - 3 \cdot P$

Let's $X = 1023_{10}\ (\text{mod}\ 23) = (1111111111)_2\ (\text{mod}\ 23)$

4) *X (mod P)*, **where** $P = 2^n$

$$X \left(\bmod\ 2^n\right) = \left(x_\delta,\ x_{\delta-1}, \ldots, x_n, x_{n-1}, \ldots, x_1\right) \left(\bmod\ 2^n\right) =$$

$$= \left(x_n, x_{n-1}, \ldots, x_1\right) \left(\bmod\ 2^n\right)$$

$$x_1, x_2, x_3, x_4, \ldots \in \{0,1\}$$

Example. ***X (mod P)***, where $X = (0111110101\ 1100111101\ 101)$ and $P = 16$

$$X = (0111110101\ 1100111101\ 101)(\bmod\ 16) = (1101) = 14$$

Using one of the considered techniques, calculate:

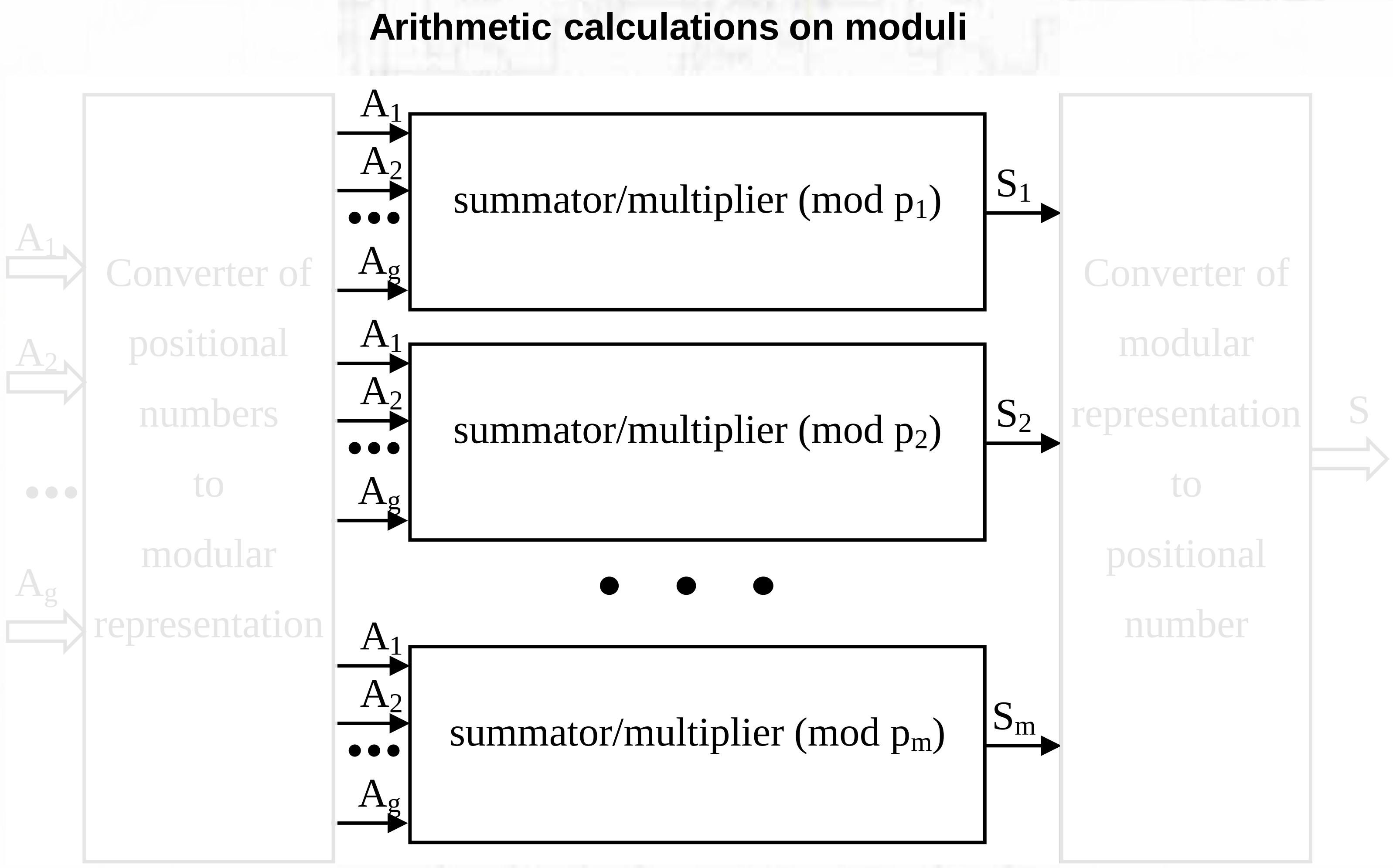
1) $65536 \pmod{2^3} = 0$

2) $65536 \pmod{2^3 - 1} = 2$

3) $(1010101010101010101) \pmod{2^3 - 1} = 1$

4) $(1010101010101010101) \pmod{2^3 - 1} = 1$ with technique 3)

Arithmetic calculations in RNS



Standard approach of arithmetic calculations in RNS includes

1) arithmetic calculations (*A·B=R*, *A+B=S*, and etc., where *A* and *B* vary from *0* to *P-1*);

2) modulus function calculation (*R (mod P)*, *S (mod P)*, and etc.)

Example. *A·B=R (mod 7)*, hence *A* and *B* vary from 0 to 6. Lets *A=5* and *B=6*.

1) $5 \cdot 6 = 30$

2) $30_{10} (\text{mod } 7) = (11110)_2 (\text{mod } 7) =$

$= ((011) + (110)) (\text{mod } 7) = (1001) (\text{mod } 7) =$

$= ((001) + (001)) (\text{mod } 7) =$

$= (010)_2 (\text{mod } 7) = 2_{10} (\text{mod } 7)$

What is about $P \approx 2^{300}$?

1) $A \cdot B = R \le 2^{600}$

2) $R \pmod{2^{300}}$

or Montgomery and "a-la Montgomery" multiplication:

Example.

$$(5 \cdot 6) \pmod 7 =$$
$$= ((101)_2 \cdot (110)_2) \pmod 7 = ((2^2 + 2^0) \cdot (2^2 + 2^1)) \pmod 7 =$$
$$= 2^4 \pmod 7 + 2^3 \pmod 7 + 2^2 \pmod 7 + 2^1 \pmod 7 = (2 + 1 + 4 + 2) \pmod 7 =$$
$$= 9 \pmod 7 = (1001) \pmod 7 = 2^3 \pmod 7 + 2^0 \pmod 7 = (1 + 1) \pmod 7$$

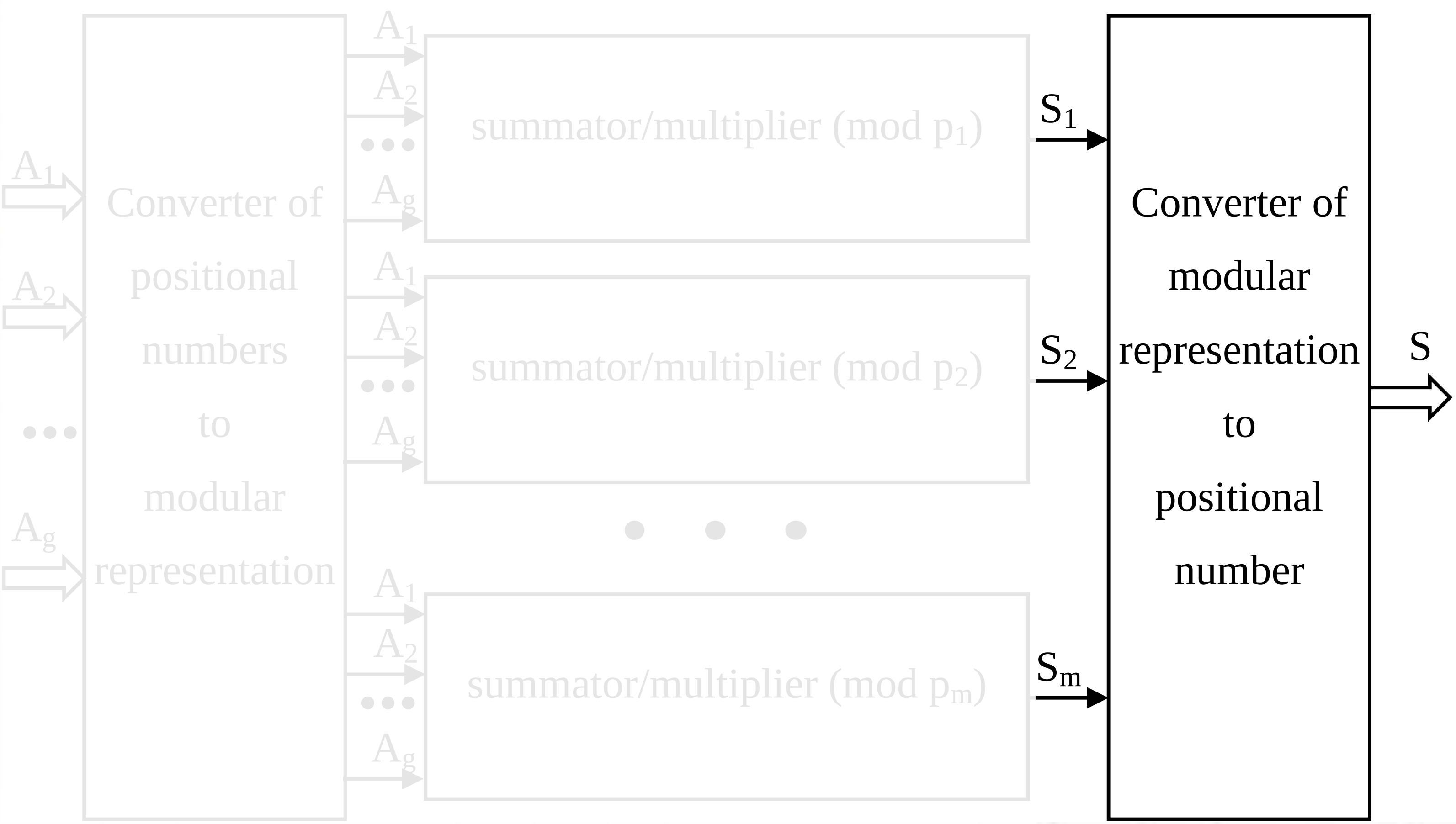
Exercises

1) How many rows and columns in the truth table of system of Boolean functions, which represents $A + B = R \ (mod\ 15)$?

2) How many rows and columns in the truth table of system of Boolean functions, which represents $A * B = R \ (mod\ 17)$?

Backward conversion

$$S = S_1 \cdot Y_1 + S_2 \cdot Y_2 + \ldots + S_m \cdot Y_m - r \cdot P$$



$$S = S_1 \cdot Y_1 + S_2 \cdot Y_2 + ... + S_m \cdot Y_m - r \cdot P$$

$$\frac{Y_i}{p_i} = 1 \pmod{p_i} \qquad Y_i = \left(\frac{P}{p_i}\right) k_i$$

1) multiplication by a big number;

2) big numbers summation;

3) comparison

Example.

$A \cdot B$ in RNS with moduli set $P = \{p_1, p_2, p_3\} = \{31, 32, 33\}$

$$R = s_1 \cdot 16864 + s_2 \cdot 31713 + s_3 \cdot 16896 - r \cdot 32736$$

Exercises

Multiply *129 * 103* in RNS with moduli set {11, 13, 15, 16}?

- Synopsys – executes X mod P;
- Xilinx (ISE, Vivado) – implementation IP-blocks;
- LeonardoSpectrum (Mentor Graphics) – allows to use custom libraries;
- and etc.

Special sets of moduli applied in RNS

Moduli Set	Year
$\{2^n-1, 2^n, 2^n+1\}$	1967
$\{2n-1, 2n, 2n+1\}$	1992
$\{2^{2n}+1, 2^n+1, 2^n-1\}$	1997
$\{2^n-1, 2^n, 2^{n-1}-1\}$	1998
$\{2^n-1, 2^n, 2^{n+1}-1\}$	1999
$\{2^n-1, 2^n, 2^{2n+1}-1\}$	2008
$\{2^{2n}-1, 2^n, 2^{2n}+1\}$	2008
$\{2^\alpha, 2^\beta-1, 2^\beta+1\}$	2008
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$	1999
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$	2000
$\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$	2003
$\{2^n-1, 2^n+1, 2^n-3, 2^n+3\}$	2004
$\{2^n-1, 2^n+1, 2^{2n}-2, 2^{2n+1}-3\}$	2008
$\{2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1\}$	2010
$\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$	2010
$\{2^n-1, 2^n+1, 2^{2n}, 2^{2n+1}-1\}$	2010
$\{2^k, 2^n-1, 2^n+1, 2^{n+1}+1\}$	2014
$\{2^k, 2^n-1, 2^n+1, 2^{n-1}-1\}$	2014
$\{2^n-1, 2^n, 2^n+1, 2^n-2^{(n+1)/2}+1, 2^n+2^{(n+1)/2}+1\}$	2005
$\{2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1\}$	2007
$\{2^{n/2}-1, 2^n, 2^{n/2}+1, 2^n+1, 2^{2n-1}-1\}$	2009
$\{2^n-1, 2^n, 2^n+1, 2^n-2^{(n+1)/2}+1, 2^n+2^{(n+1)/2}+1, 2^{n\pm1}+1\}$	2013
$\{2^n-1, 2^{n+\beta}, 2^n+1, 2^n-2^{(n+1)/2}+1, 2^n+2^{(n+1)/2}+1, 2^{n\pm1}+1\}$	2013

Five moduli set

In order to calculate $A \cdot B$ **, where** $A, B \le 2^{739}$ **,**

the average bit-range of 5 moduli sets is 300 bits, i.e.

$$p_1 \approx p_2 \approx p_3 \approx p_4 \approx p_5 \approx 2^{300}$$

Moduli set for $A \cdot B$, where A and B are 739 bits

P = {1021 1019 1013 1009 997 991 983 977 971 967 961 953
947 941 937 929 919 911 907 887 883 881 877 863 859 857
853 841 839 829 827 823 821 811 809 797 787 773 769 761
757 751 743 739 733 729 727 719 709 701 691 683 677 673
661 659 653 647 643 641 631 625 619 617 613 607 601 599
593 587 577 571 569 563 557 547 541 529 523 521 512 509
503 499 491 487 479 467 463 461 457 449 443 439 433 431
421 419 409 401 397 389 383 379 373 367 361 359 353 349
347 343 337 331 317 313 311 307 293 289 283 281 277 271
269 263 257 251 241 239 233 229 227 223 211 199 197 193
191 181 179 173 169 167 163 157 151 149 139 137 131 127
121 113 109 107 103 101 97 89 83 79 73 71 67 61 59
53 47 43 41 37}

$|P|=172$ $P=2^{1478}$

It is assumed, that:

– the main feature is the high speed processing (it is achieved with hundreds bits numbers);

– independence of calculation under each modulo;

– flexibility of layout;

– small power consumption;

– reliability

Problems:

– unknown an *efficient* approach of hardware realization for an arbitrary modulo *P*

– no IP-blocks and no hardware libraries for RNS system realization;

– slow speed converters to/from RNS for non special sets of moduli

Residue Number Systems: Algorithms and Architectures

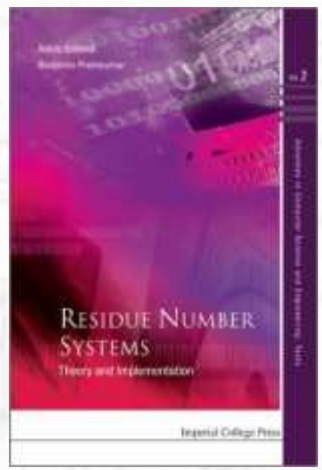
Kluwer Academic Publishers, 2002

Digital arithmetic

Morgan Kaufmann Publishers, 2004

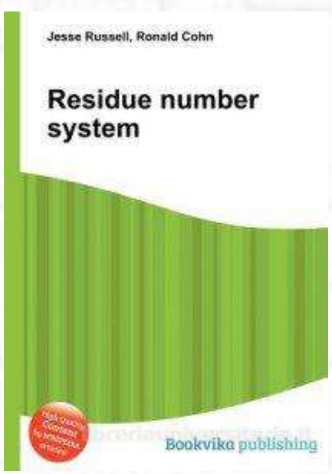
Residue Number Systems: Theory and Implementation

Imperial College Press, 2007



Residue Number System

Bookvika Publishing, 2012



Finite precision number systems and arithmetic

Cambridge University Press, 2010