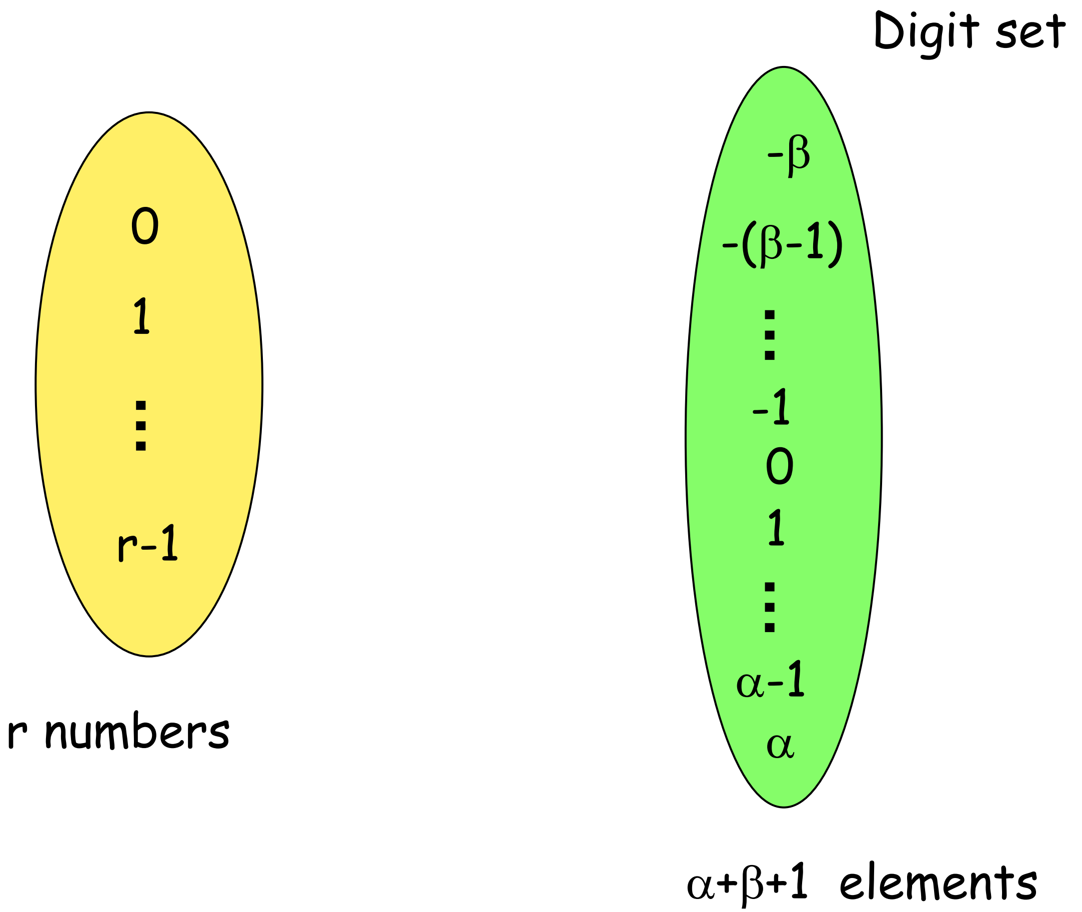
VLSI Signal Processing

Lecture 9 Redundant Arithmetic

Redundant?

- A non-redundant radix-r number has digits from the set {0,1,…,r-1} and all numbers can be represented in a unique way
- A radix-r redundant signed-digit number system is based on digit set $S \equiv \{-\beta,\dots,-1,0,1,\dots,\alpha\}$, where $1 \le \beta,\alpha \le r-1$
- The digit set S contains more than r values ➔ multiple representations for any number in signed digit format. Hence, the name redundant
- A symmetric signed digit has $\alpha=\beta$.
- Carry-free addition is one of the most attractive properties of redundant signed-digit numbers. This allows most significant digit (MSD) first redundant arithmetic, also called on-line arithmetic (division..)

1-1 onto Mapping?



Redundant Number Representations

- A symmetric signed-digit representation uses the digit set $D_{\langle r,\alpha\rangle}=\{-\alpha,\ldots,-1,0,1,\ldots,\alpha\}$, where r is the radix and α the largest digit in the set.
- A number in this representation is written as:

$$X_{\langle r,\alpha\rangle}=x_{W-1}\circ x_{W-2}\ldots x_1x_0=\Sigma\, x_{W-1-i}r^i$$

- The sign of the number X is given by the sign of the most significant non-zero digit
- If $2\alpha+1<r$, the digit set $D_{\langle r,\alpha\rangle}$ is incomplete (i.e. some numbers cannot be represented)
- If $2\alpha+1=r$, the digit set $D_{\langle r,\alpha\rangle}$ is complete but not redundant
- If $2\alpha+1>r$, the digit set $D_{\langle r,\alpha\rangle}$ is redundant

Example

- Signed-Digit Numbers $X_{\langle r,\alpha\rangle}=x_{W-1}\circ x_{W-2}\ldots x_1 x_0=\sum x_{W-1-i}r^i$

$$\{-1, 0, 1\}$$

Represent $-\frac{1}{4}$

1	1/2	1/4
0	0	-1
0	-1	1
-1	1	1

Canonic Signed Digit (CSD)

3 different

Redundant

Needs 2 bits since three numbers (-1,0,1)!

Review of Canonical Signed Digit

- Minimum number of non-zero bits
- A sequence of ones can be replaced with
  - A "-1" at the least significant position of the sequence
  - A "1" at the position to the left of the most significant position of the sequence
  - Zeros between the "1" and the "-1"

- Save more than 2/3 of the adder cells at an average

Redundant Characteristics

- Redundant factor
  - A measure of the redundancy of a symmetric signed-digit representations : $\rho = \alpha/(r-1)$

Digit Set $D_{\langle r,\alpha \rangle}$	α	Redundancy Factor ρ
Incomplete	$< (r-1)/2$	$< \frac{1}{2}$
Complete but non-redundant	$= (r-1)/2$	$= \frac{1}{2}$
Redundant	$\geq \lceil r/2 \rceil$	$> \frac{1}{2}$
Minimally redundant	$= \lceil r/2 \rceil$	$> \frac{1}{2}$ and < 1
Maximally redundant	$= r - 1$	$= 1$
Over-redundant	$> r - 1$	> 1

Redundancy Characteristic Example

Radix 4 with digit set {-2,-1,0,1,2}

$$\rho = \frac{\alpha}{r-1} = \frac{2}{3}$$ ➡ **Minimally redundant**

Radix 4 with digit set {-3,-2,-1,0,1,2,3}

$$\rho = \frac{\alpha}{r-1} = \frac{3}{3}$$ ➡ **Maximally redundant**

Carry-Free ?

- Redundant number representations limit the carry propagation to a few bit-positions, which is usually independent of the wordlength W.
- + and -

+

- Carry-free arithmetic
  - independent of W
- MSD – first

1	1/2	1/4
0	0	-1
0	-1	1
-1	1	1

-

- Hard to find the sign
- More bits
- Conversion
- Not good for non-arithmetic

Radix-2 Signed-Digit Number

Each signed digit is represented by 2 bits

$\{-1, 0, 1\}$ **3 values** → **2 bits representation**

x^+	x^-	x
0	0	0
0	1	-1
1	0	1
1	1	0

Redundant

$$X = X^+ - X^-$$

Two unsigned binary numbers

Redundant Arithmetic

- $X_{\langle r,\alpha\rangle} = X^{+} - X^{-}$, $Y_{\langle r,\alpha\rangle} = Y^{+} - Y^{-}$

- Hybrid radix-r addition $X_{\langle r,\alpha\rangle} + Y$
- Hybrid radix-r subtraction $X_{\langle r,\alpha\rangle} - Y$

- A signed-digit addition/subtraction can then be viewed as a concatenation of one hybrid addition/subtraction and one hybrid subtraction/addition

Hybrid Radix-2 Addition

Signed digit number $\{-1, 0, 1\}$

$$S_{<2.1>} = X_{<2.1>} + Y \leftarrow \text{unsigned}$$

where, $X_{<r.\alpha>} = x_{W-1}.x_{W-2}x_{W-3}\ldots x_0$, $Y = y_{W-1}.y_{W-2}y_{W-3}\ldots y_0$. The addition is carried out in two steps :

1. The 1st step is carried out in parallel for all the bit positions. An intermediate sum $p_i = x_i + y_i$ is computed, which lies in the range $\{\bar{1}, 0, 1, 2\}$. The addition is expressed as:

$$x_i + y_i = 2t_i + u_i,$$

   where t_i is the transfer digit and has value 0 or 1, and is denoted as t_i^+; u_i is the interim sum and has value either 1 or 0 and is denoted as $-u_i^-$. t_{-1} is assigned the value of 0.

2. The sum digits s_i are formed as follows:

$$s_i = t_{i-1}^+ - u_i^-$$

Hybrid Radix-2 Addition

Digit	Radix 2 Digit Set	Binary Code
x_i	$\{\bar{1}, 0, 1\}$	$x_i^+ - x_i^-$
y_i	$\{0, 1\}$	y_i^+
$p_i = x_i + y_i$	$\{\bar{1}, 0, 1, 2\}$	$2t_i + u_i$
u_i	$\{\bar{1}, 0\}$	$-u_i^-$
t_i	$\{0, 1\}$	t_i^+
$s_i = u_i + t_{i-1}$	$\{\bar{1}, 0, 1\}$	$s_i^+ - s_i^-$

Plus-Plus-Minus

Pass to the next level

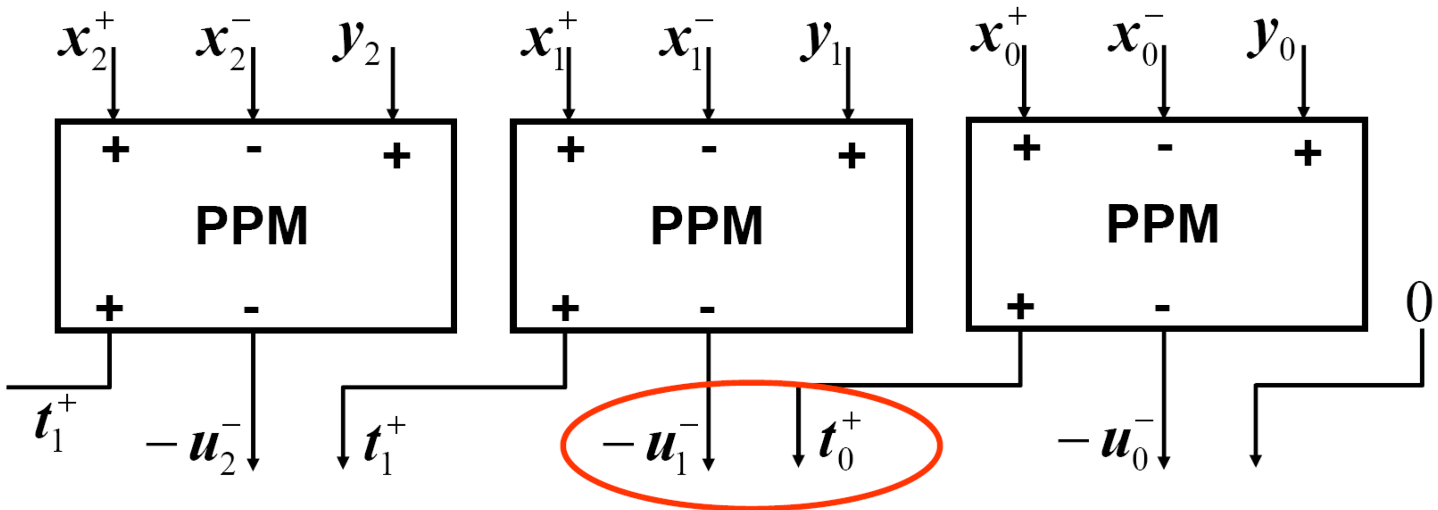
$$s_i = t_{i-1}^+ - u_i^-$$

$p_i = x_i + y_i = 2t_i + u_i$

Binary code

Hybrid Radix-2 Adder

- PPM: Plus-Plus-Minus adder (Full adder)

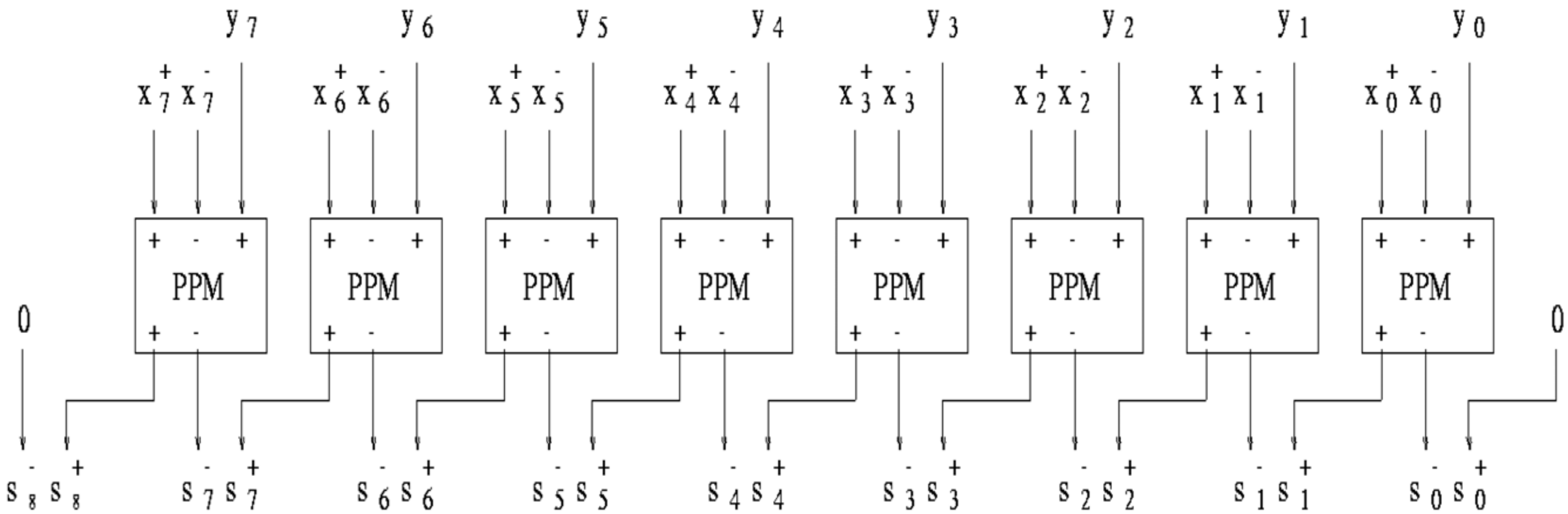


$$s_1 = s_1^+ - s_1^- = t_0^+ - u_1^-$$

No carry!

Example

- Eight-digit hybrid radix-2 adder



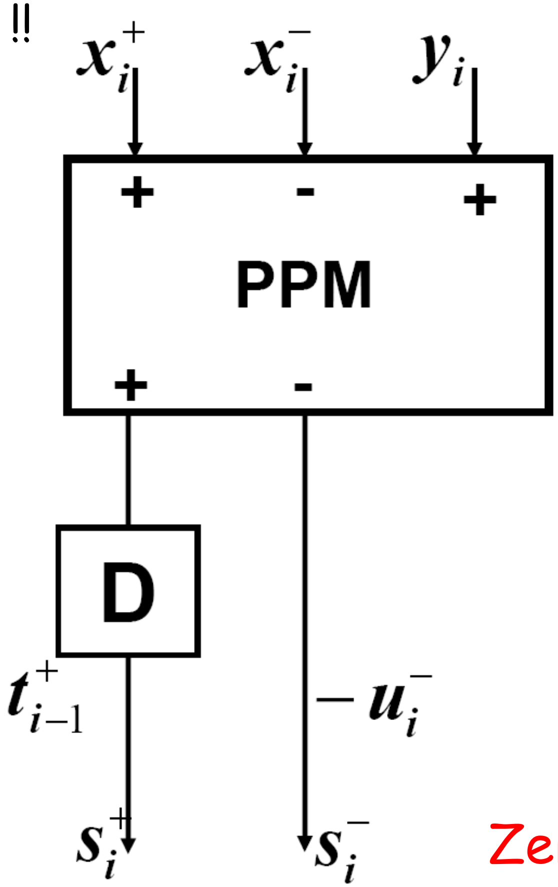
The sum has 9 digits !!

Digit-Serial Arithmetic, LSD

By Folding !!

Transfer digit delayed

Least Significant Digit (LSD) First



Serial I/O

Zero-latency !!

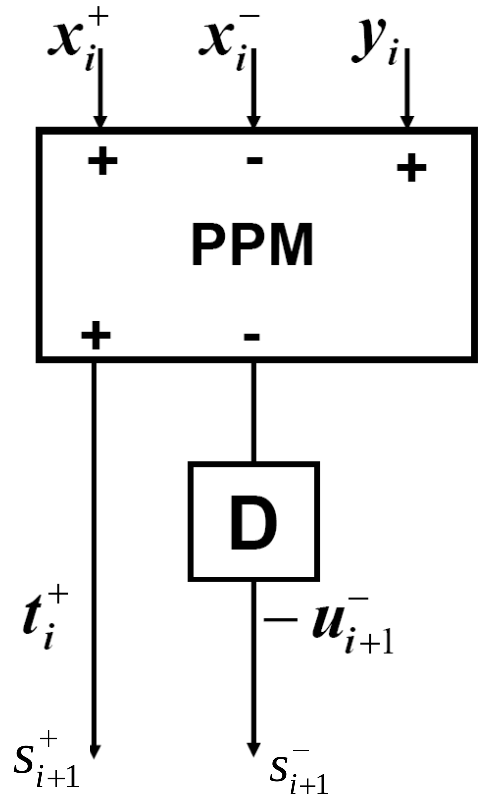
One 8-digit addition takes 9 clock cycles

Digit-Serial Arithmetic, MSD

interim sum digit delayed

Most Significant Digit (MSD) First

Good for e.g. Division



One clock cycle latency !!

Note: a zero digit needs to be inserted between 2 consecutive input operands

Hybrid Radix-2 Subtraction

Signed digit number $\{-1, 0, 1\}$

$$S_{<2.1>} = X_{<2.1>} - Y \leftarrow \text{unsigned}$$

where, $X_{<r.\alpha>} = x_{W-1}.x_{W-2}x_{W-3}...x_0$, $Y = y_{W-1}.y_{W-2}y_{W-3}...y_0$. The subtraction is carried out in two steps :

1. The 1st step is carried out in parallel for all the bit positions. An intermediate difference $p_i = x_i - y_i$ is computed, which lies in the range $\{\bar{2}, \bar{1}, 0, 1\}$. The addition is expressed as:

   $$x_i - y_i = 2t_i + u_i,$$

   where t_i is the transfer digit and has value 1 or 0, and is denoted as $-t_i^-$; u_i is the interim sum and has value either 0 or 1 and is denoted as u_i^+. t_{-1} is assigned the value of 0.
2. The sum digits s_i are formed as follows:

   $$s_i = -t_{i-1}^- + u_i^+$$

DEPT. OF ELECTRONICS ENGINEERING & INST. OF ELECTRONICS NCTU

Hybrid Radix-2 Subtraction

Digit	Radix 2 Digit Set	Binary Code
x_i	$\{\bar{1}, 0, 1\}$	$x_i^+ - x_i^-$
y_i	$\{0, 1\}$	y_i^-
$p_i = x_i - y_i$	$\{\bar{2}, \bar{1}, 0, 1\}$	$2t_i + u_i$
u_i	$\{0, 1\}$	u_i^+
t_i	$\{\bar{1}, 0\}$	$-t_i^-$
$s_i = u_i + t_{i-1}$	$\{\bar{1}, 0, 1\}$	$s_i^+ - s_i^-$

Minus-Minus-Plus

Pass to the next level

$$s_i = -t_{i-1}^- + u_i^+$$

Use negative transfer digit !!

Remarks

- {-1, 0, 1} − {0, 1} = {-2,-1, 0, 1}

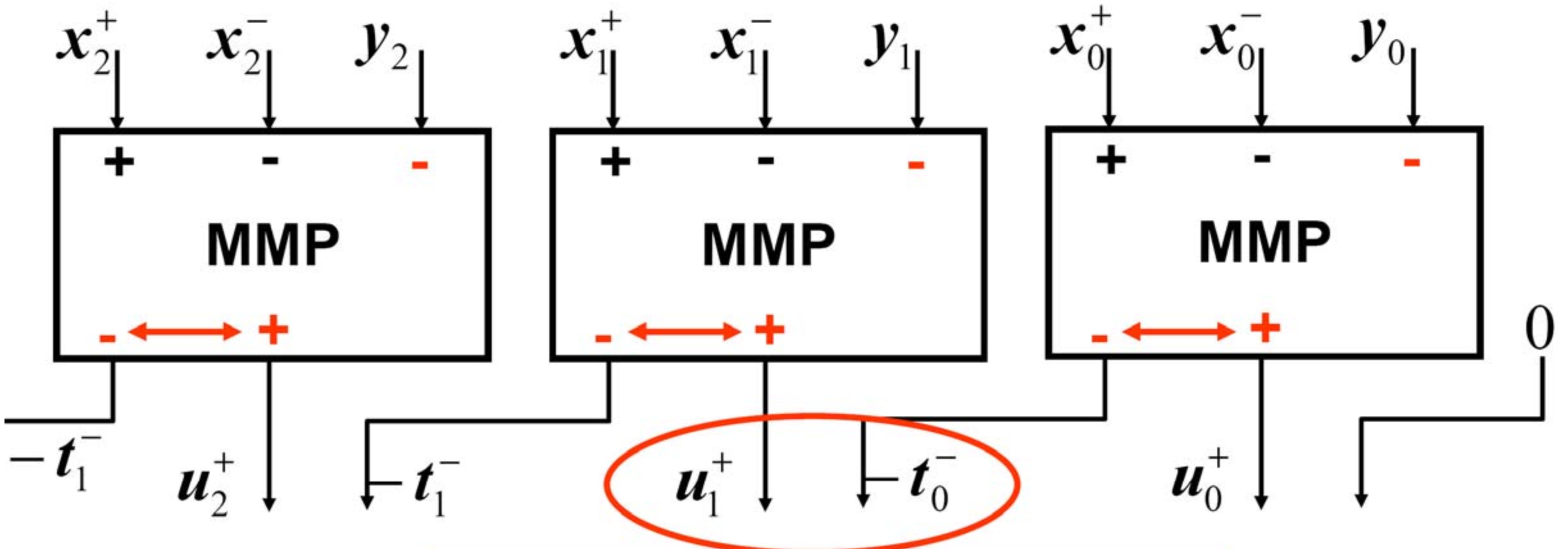
$x_i = x_i^+ - x_i^-$ $\qquad\qquad$ y_i

$$\begin{cases} \mathbf{1} = \boldsymbol{u}_i^+ \\ \mathbf{0} \\ -\mathbf{1} = -\mathbf{2}\boldsymbol{t}_i^- + \boldsymbol{u}_i^+ \\ -\mathbf{2} = -\mathbf{2}\boldsymbol{t}_i^- \end{cases}$$

Pass to the next level

Hybrid Radix-2 Subtractor

- MMP: Minus-Minus-Plus adder (Full adder)

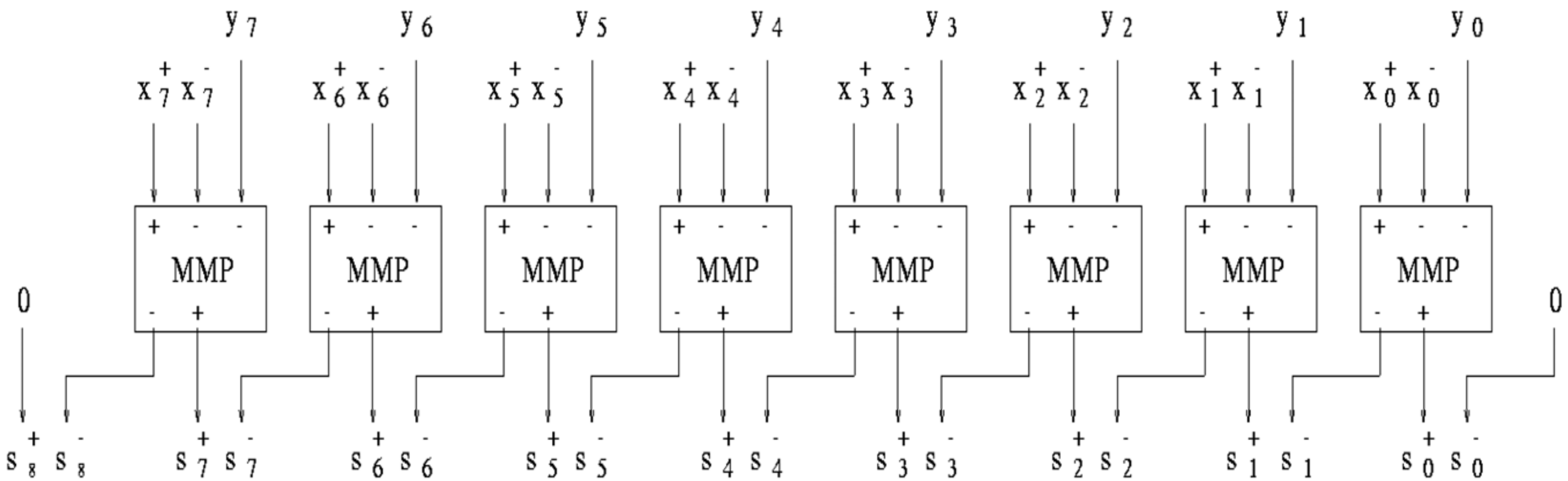


$$\boxed{s_1 = s_1^+ - s_1^- = -t_0^- + u_1^+}$$

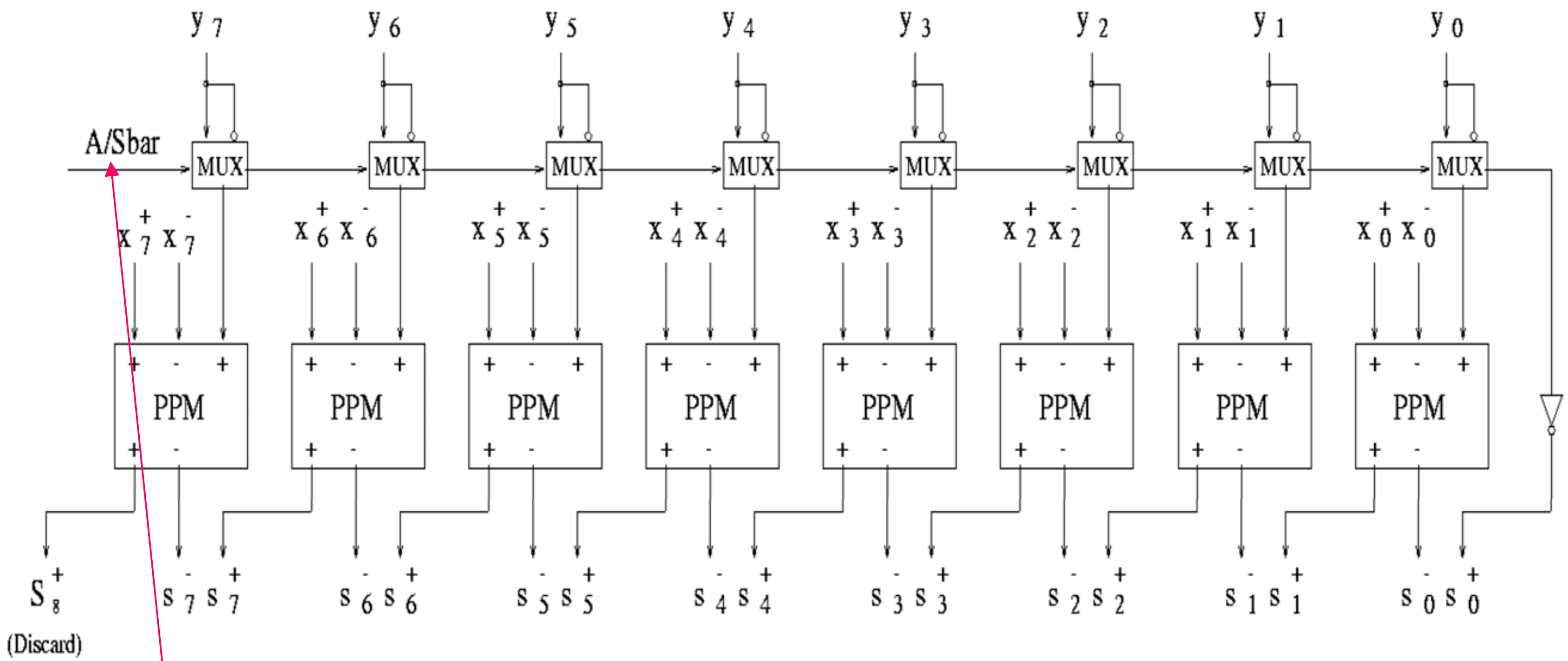
DEPT. OF ELECTRONICS ENGINEERING & INST. OF ELECTRONICS NCTU

Example

- Eight-digit hybrid radix-2 subtractor



Hybrid Radix-2 Addition/Subtraction



A/Sbar =1 for addition, A/Sbar =0 for subtraction

Signed-Binary Digit (SBD) Arithmetic

- $Y_{\langle r.\alpha \rangle} = Y^{+} - Y^{-}$, is a signed digit number, where Y^{+} and Y^{-} are from the digit set $\{0, 1, \ldots, \alpha\}$.
- A signed digit number is thus subtraction of 2 unsigned conventional numbers.
- Signed addition is given by:

$$S_{\langle r.\alpha \rangle} = X_{\langle r.\alpha \rangle} + Y_{\langle r.\alpha \rangle} = X_{\langle r.\alpha \rangle} + Y^{+} - Y^{-},$$

$$\Rightarrow S1_{\langle r.\alpha \rangle} = X_{\langle r.\alpha \rangle} + Y^{+},$$

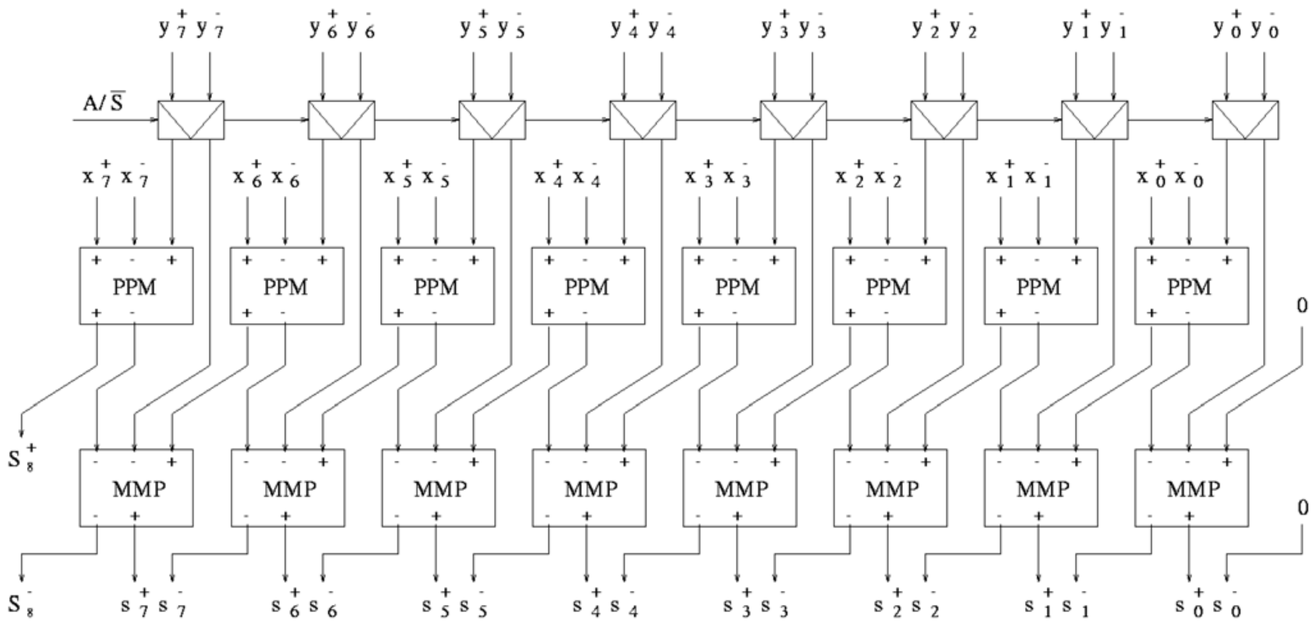
$$S_{\langle r.\alpha \rangle} = S1_{\langle r.\alpha \rangle} - Y^{-}$$

- Digit serial SBD adders can be derived by folding the digit parallel adders in both lsd-first and msd-first modes.
- LSD-first adders have zero latency and msd-first adders have latency of 2 clock cycles.

Signed Binary Digit Arithmetic

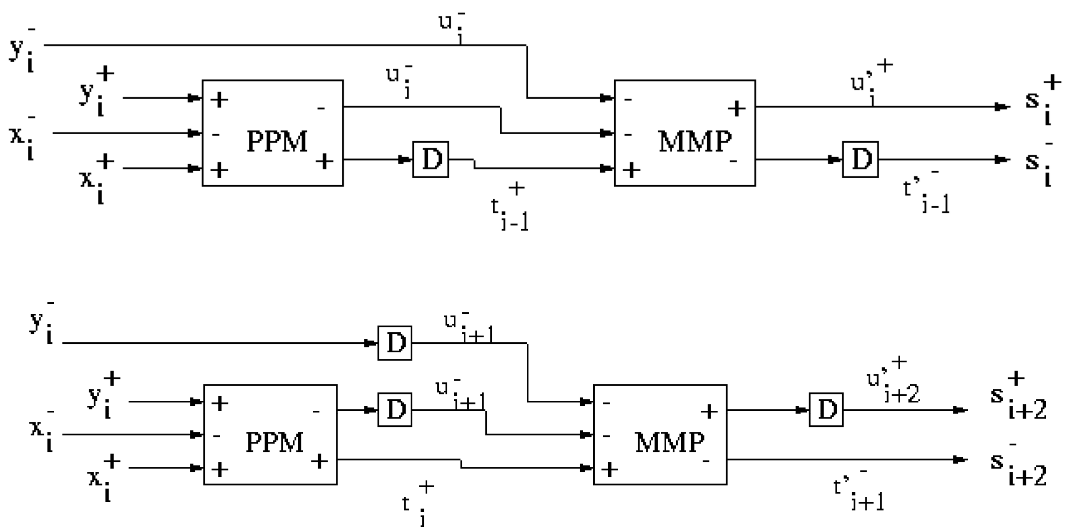
- $Y_{<r,\alpha>} = Y^{+} - Y^{-}$, is a signed digit number, where Y^{+} and Y^{-} are (unsigned conventional numbers) from the digit set $\{0,1,\ldots,\alpha\}$.
- Signed addition is given by

$$S_{<r,\alpha>}=X_{<r,\alpha>} + Y_{<r,\alpha>}=(X_{<r,\alpha>}+Y^{+})-Y^{-}$$



Remarks

- Signed-digit addition can be viewed as a concatenation of one hybrid addition and one hybrid subtraction.
- Digit-serial redundant adders can be derived by folding methodology (LSD-first, or MSD-first)
- LSD-first adders have zero latency, while MSD-first adders have 2 clock cycles latency.



Hybrid Radix-4 Addition

- Higher order radices can be employed to reduce the number of iteration cycles.
- Maximally redundant hybrid radix-4 addition (MRHY4A) considers the numbers based on digit set $D_{\langle 4,3 \rangle}=\{-3,-2,-1,0,1,2,3\}$, $\rho=1$
- Minimally redundant hybrid radix-4 addition (mrHY4A) considers the numbers based on digit set $D_{\langle 4,2 \rangle}=\{-2,-1,0,1,2\}$, $\rho=2/3$

MRHY4A

$$S_{\langle 4.3\rangle} = X_{\langle 4.3\rangle} - Y_4 \longleftarrow \text{unsigned}$$

- The first step computes:

  $$x_i + y_i = 4t_i + u_i$$

  Replacing the respective binary codes from the table the following is obtained :

  $$(2x_i^{+2} - 2x_i^{-2} + 2y_i^{+2}) + x_i^{+} - x_i^{-} + y_i^{+} = 4t_i^{+} + 2u_i^{+2} - 2u_i^{-2} - u_i^{-}$$

  A MRHY4A cell consisting of two PPM adders is used to compute the above.
- Step 2 computes computes $s_i = t_{i-1} + u_i$. Replacing s_i, u_i, and t_{i-1} by corresponding binary codes leads to $s_i^{+2} = u_i^{+2}$, $s_i^{-2} = u_i^{-2}$, $s_i^{+}=t_{i-1}^{+}$ and $s_i^{-} = u_i^{-}$.

Digit Sets in MRHR4A

$D_{\langle 4,3 \rangle}=\{-3,-2,-1,0,1,2,3\}$,

Digit	Radix 4 Digit Set	Binary Code
x_i	$\{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$	$2x_i^{+2} - 2x_i^{-2} + x_i^{+} - x_i^{-}$
y_i	$\{0, 1, 2, 3\}$	$2y_i^{+2} + y_i^{+}$
$p_i = x_i + y_i$	$\{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3, 4, 5, 6\}$	$4t_i + u_i$
u_i	$\{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2\}$	$2u_i^{+2} - 2u_i^{-2} - u_i^{-}$
t_i	$\{0, 1\}$	t_i^{+}
$s_i = u_i + t_{i-1}$	$\{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$	$2s_i^{+2} - 2s_i^{-2} + s_i^{+} - s_i^{-}$

PPM

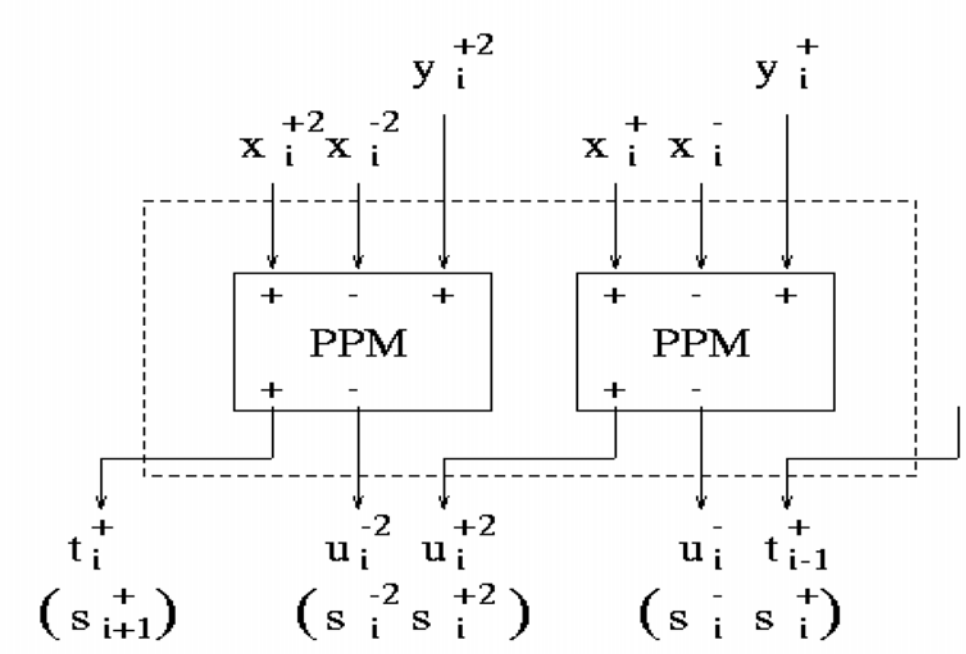
Pass to the next level

$$x_i + y_i = p_i = 4t_i + u_i$$

$$s_i = u_i + t_{i-1}, \quad \text{where } s_i^{+2} = u_i^{+2}, s_i^{-2} = u_i^{-2}, s_i^{+} = t_{i-1}^{+}, s_i^{-} = u_i^{-}$$

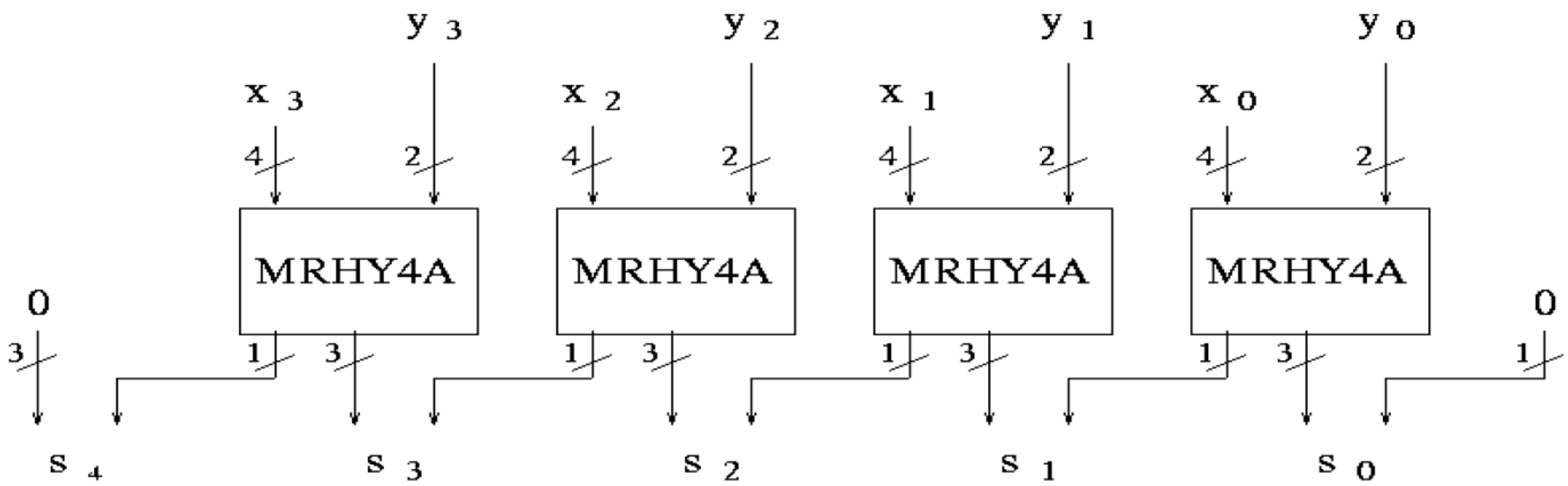
MRHY4A Adder Cell

Step 1



2PPM (full) adders that perform the 2 grouped additions in parallel and reduce the number of bits from 6 to 4 with the weights.

4-Digit MRHY4A



mrHY4A

$$S_{\langle 4.2\rangle} = X_{\langle 4.2\rangle} + Y_4 \longleftarrow \text{unsigned}$$

- The first step computes:

$$x_i + y_i = 4t_i + u_i$$

Replacing the respective binary codes from the table the following is obtained :

$$(-2x_i^{-2} + 2y_i^{+2}) + (x_i^{+} + x_i^{++} + y_i^{+}) = 4t_i^{+} - 2u_i^{-2} + u_i^{+}$$

A mrHY4A cell consisting of one PPM adder and a full adder is used to compute the above.

- Step 2 computes computes $s_i = t_{i-1} + u_i$. Replacing s_i, u_i, and t_{i-1} by corresponding binary codes leads to $s_i^{-2} = u_i^{-2}$, $s_i^{++} = t_{i-1}^{+}$ and $s_i^{+} = u_i^{+}$.

Digit Sets in mrHY4A

$D_{<4,2>}=\{-2,-1,0,1,2\}$ The digit number is represented by 3 bits !!

Digit	Radix 4 Digit Set	Binary Code
x_i	$\{\bar{2},\bar{1},0,1,2\}$	$-2x_i^{-2}+x_i^{+}+x_i^{++}$
y_i	$\{0,1,2,3\}$	$2y_i^{+2}+y_i^{+}$
$p_i=x_i+y_i$	$\{\bar{2},\bar{1},0,1,2,3,4,5\}$	$4t_i+u_i$
u_i	$\{\bar{2},\bar{1},0,1\}$	$-2u_i^{-2}+u_i^{+}$
t_i	$\{0,1\}$	t_i^{+}
$s_i=u_i+t_{i-1}$	$\{\bar{2},\bar{1},0,1,2\}$	$2s_i^{-2}+s_i^{+}+s_i^{++}$

Full adder

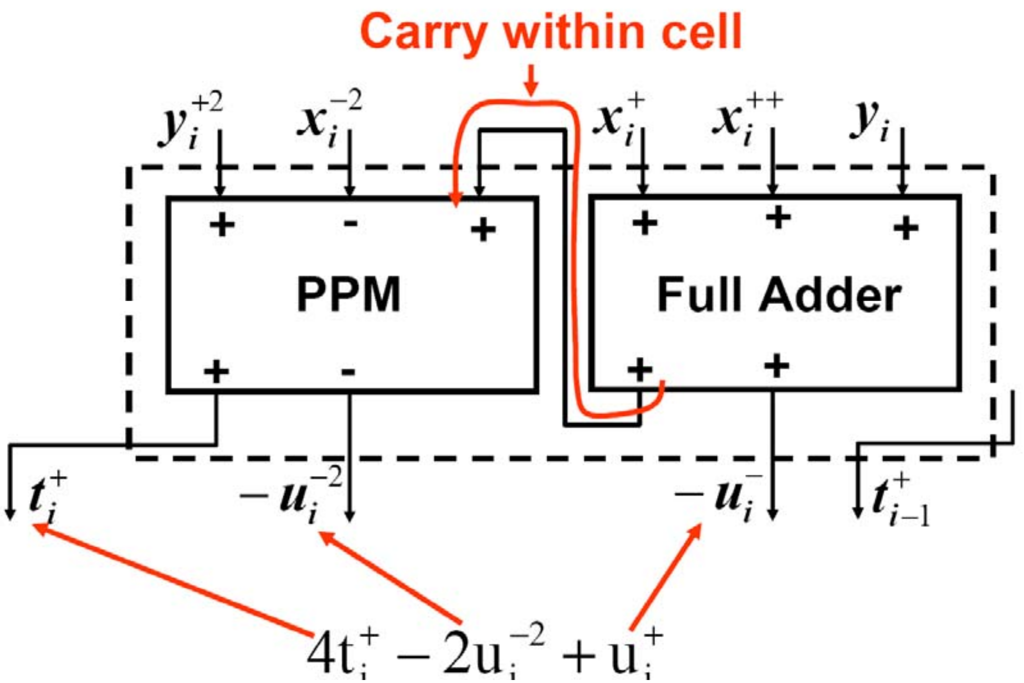
Pass to the next level

$$x_i+y_i=p_i=4t_i+u_i$$

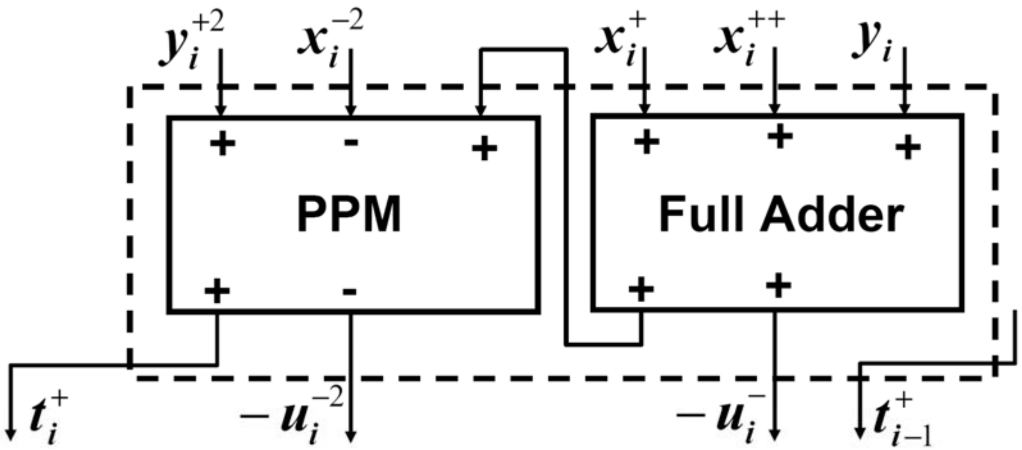
$$s_i=u_i+t_{i-1}$$

Minimally redundant Hybrid Radix-4 Adder Cell

step1



step2

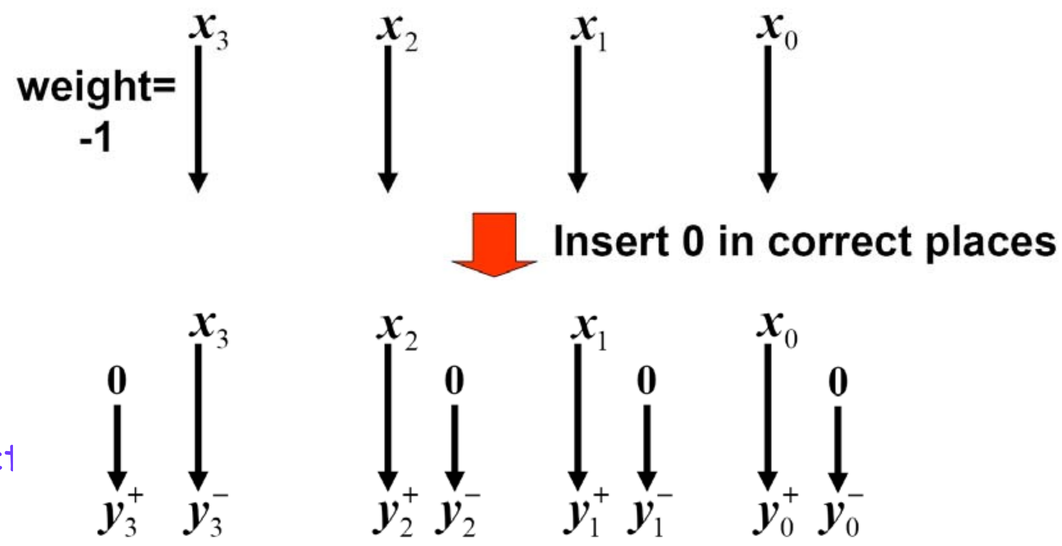


$\boxed{s_{i+1}^{++}}$ $\boxed{s_i^{-2}}$ $\boxed{\left(s_i^{+} \quad s_i^{++}\right)}$

tw)

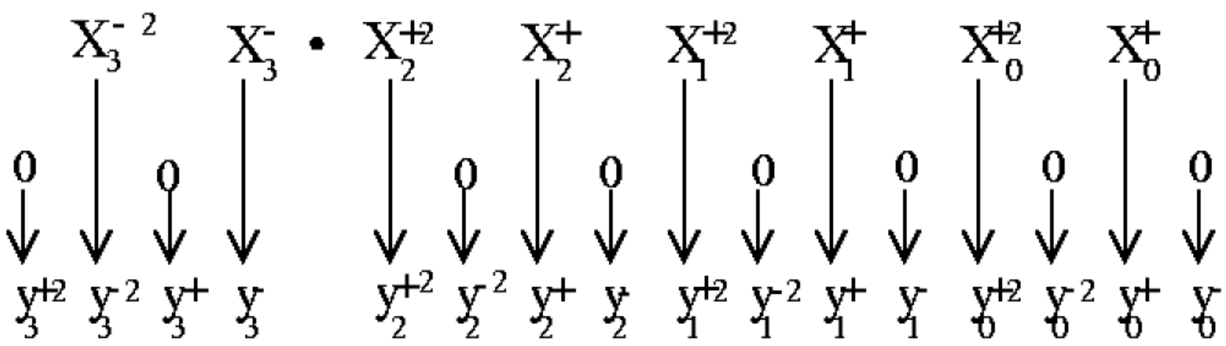
Non-Redundant to Redundant Conversion

- The non-redundant input digit set can be considered as a subset of the redundant input digit set.
- Radix-2 representation:
  - A non-redundant number $X=x_3 \circ x_2 x_1 x_0$ can be converted to a redundant number $Y=y_3 \circ y_2 y_1 y_0$, where each digit y_i is encoded as y_i^+ and y_i^- as shown below



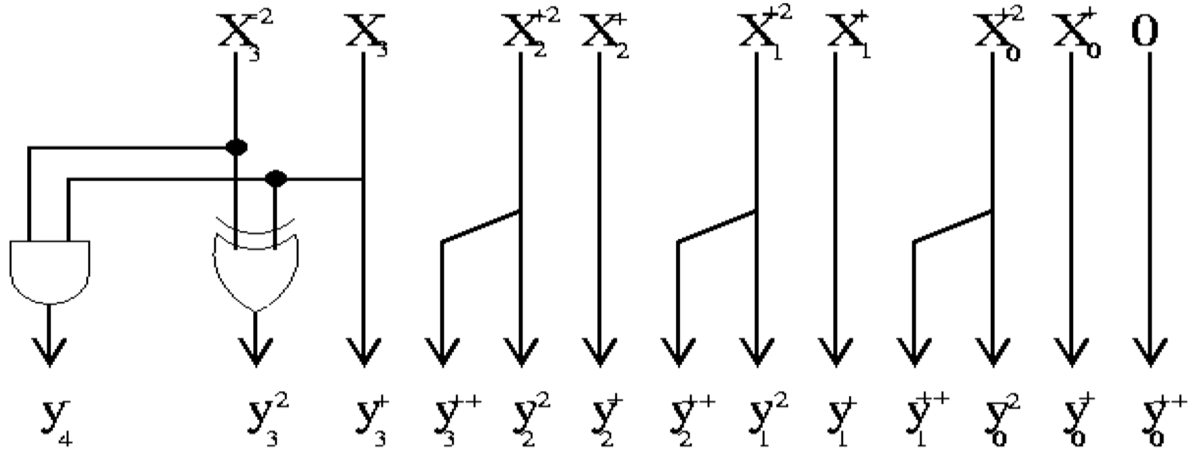
Non-Redundant to Redundant Conversion

- Radix-4 representation
  - X is a radix-4 complement number, whose digits x_i's are encoded using 2 wires as $x_i = 2x_i^{+2} + x_i^{+}$. Its corresponding maximally redundant number Y is encoded using $y_i=2y_i^{+2}-2y_i^{-2}+y_i^{+}-y_i^{-}$. The sign digit x_3 can take values -3, -2, -1, or 0, and is encoded using $x_3=-2x_3^{-2}-x_3^{-}$.



- <u>radix-4 minimally redundant number</u>: X is a radix-4 complement number, whose digits x_i are encoded using 2 wires as $x_i = 2x_i^{+2} + x_i^{+}$. Its corresponding minimally redundant number Y is encoded using $y_i = -2y_i^{-2} + y_i^{+} + y_i^{++}$. To convert radix-r number x to redundant number $y_{<r,\alpha>}$, the digits in the range $[\alpha, r - 1]$ are encoded using a transfer digit 1 and a corresponding digit $x_i - r$ where x_i is the i^{th} digit of x. Thus,

$$2x_i^{+2} + x_i^{+} = 4x_i^{+2} - 2x_i^{+2} + x_i^{+}$$
$$= y_{i+1}^{++} - 2y_i^{-2} + y_i^{+}$$



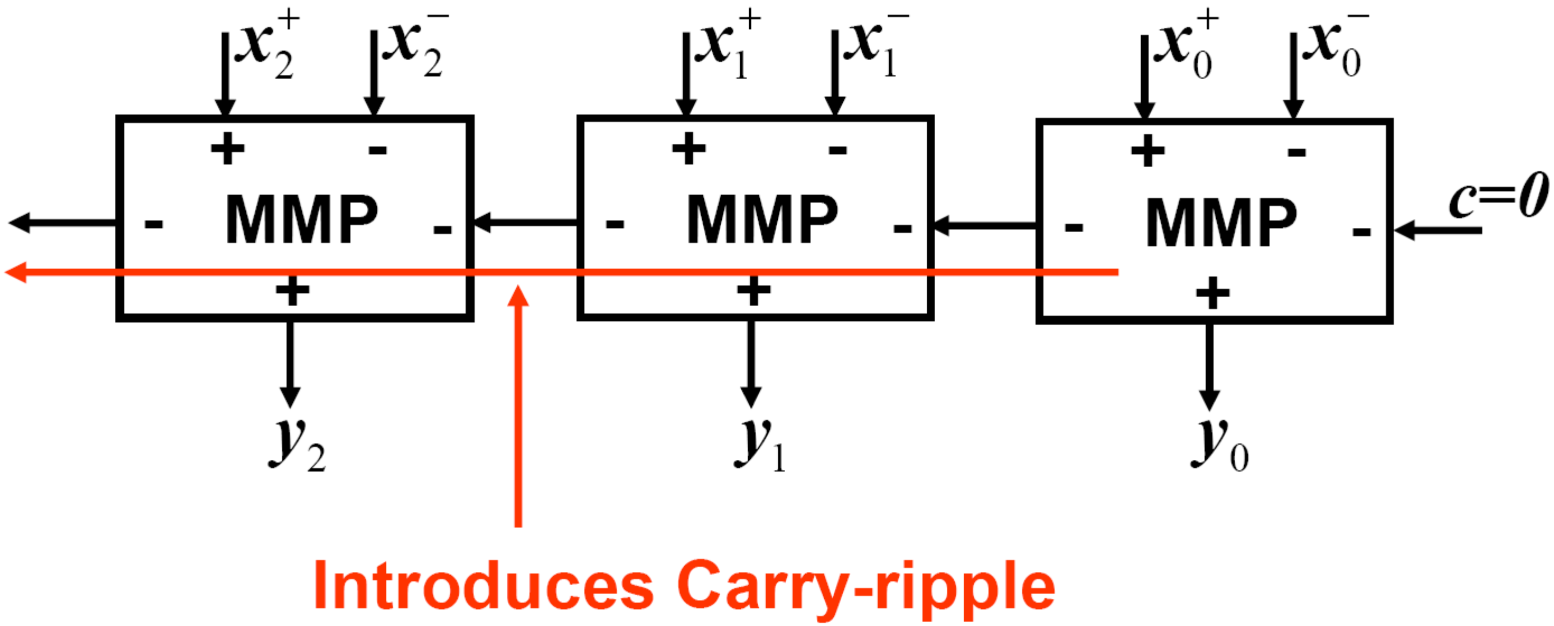
Redundant to NonRedundant Converter

- In general, it is not possible to know the value of any of nonredundant digits until the least significant redundant digit become available
- Example
  - $1000\bar{1} \longrightarrow (03333)_4$
  - $10001 \longrightarrow (10001)_4$

To change the least significant digit from -1 to 1 causes all of the preceding (more significant) digits to be changed

Redundant to Non-red Conversion

In general not possible to output ANY digit until LSD has been processed, i.e. advatage of MSD first ”removed”



If a redundant number is scanned msd-first and transformed to a nonredundant radix-4 format