Redundant Number Systems

Redundant Number Systems

- Conventional radix-r systems use [0,r-1] digit set
  - radix-10 → 0,1,2,3,4,5,6,7,8,9
- If the digit set (in radix-r system) contains more than r digits, the system is redundant
  - radix-2 → 0,1,2 or -1,0,1
  - radix-10 → 0,1,2,3,4,5,6,7,8,9,10,11,12,13
  - radix-10 → 0,1,2,3,4,5,6,7,8,9,ث,ڭ,گ,چ
  - radix-10 → -6,-5,-4,-3,-2,-1,0,1,2,3,4,5
- Conversions between Redundant Number Systems is a simple digit-serial process

Redundancy

- Redundancy – representation of numbers is not unique
- Redundancy may result from narrowing the range of represented values (e.g. 1's compl.), but number interpretation may be complex
- Redundancy may result from adopting the digit set wider than radix and the number interpretation is conventional
- Physical representation of redundant numbers may be not trivial.

Conversions

- Conversion is a carry-propagate (slow) process

	11	9	17	10	12	18	radix-10, digit set [0,18]
	11	9	17	10	12	**18**	18 = 10+8
	11	9	17	10	**13**	8	13 = 10+3
	11	9	17	**11**	3	8	11 = 10+1
	11	9	**18**	1	3	8	18 = 10+8
	11	**10**	8	1	3	8	10 = 10+0
	12	0	8	1	3	8	12 = 10+2
1	2	0	8	1	3	8	radix-10, digit set [0,9]

	11	9	17	10	12	18	radix-10, digit set [0,18]
	11	9	17	10	12	**18**	18 = 20−2
	11	9	17	10	**14**	−2	14 = 10+4
	11	9	17	**11**	4	−2	11 = 10+1
	11	9	**18**	1	4	−2	18 = 20−2
	11	**11**	−2	1	4	−2	11 = 10+1
	12	1	−2	1	4	−2	12 = 10+2
1	2	1	−2	1	4	−2	radix-10, digit set [-6,5]

Decomposition

- Redundant numbers with [0,m] digit set can be represented by two numbers of [0,n] digit sets, where m=2n
- Conversion requires ordinary addition of two such numbers with [0,n] digit set representation

	11	9	17	10	12	18	radix-10, digit set [0,18]
	9	9	9	9	9	9	
+	2	0	8	1	3	9	
1	2	0	8	1	3	8	radix-10, digit set [0,9]

- Decomposed representation is, of course, not unique, but the sum amounts to correct result

Radix-2 [0,2] digit set numbers

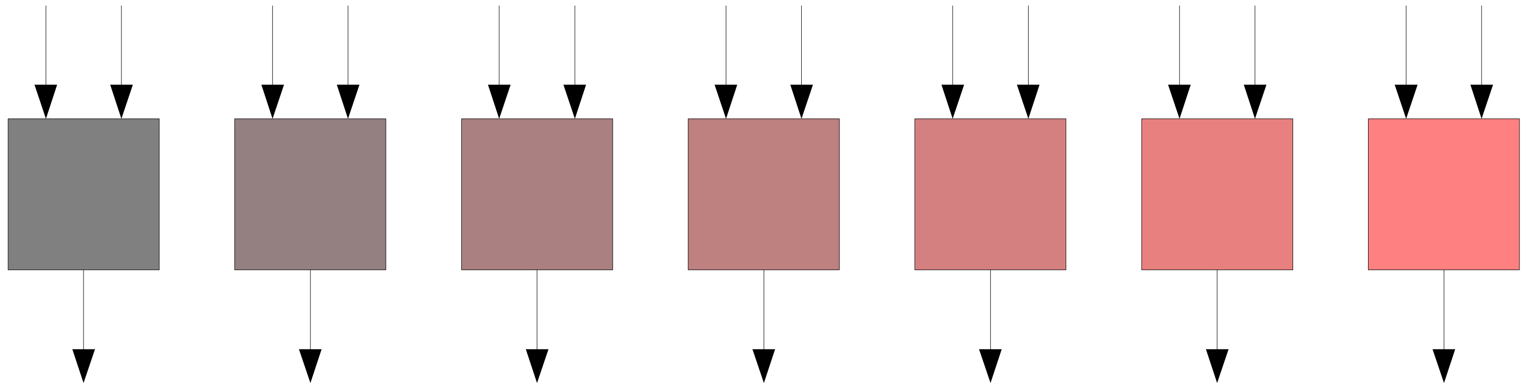
- Redundant binary numbers may be coded with bit-fields, e.g:
  0: (0,0),
  1: (0,1) or (1,0),
  2: (1,1)
- Decomposed form is convenient and efficient representation of redundant binary numbers

		1	1	2	0	2	0	radix-2, digit set [0,2]
		1	1	1	0	1	0	
+		0	0	1	0	1	0	
	1	0	0	0	1	0	0	radix-2, digit set [0,1]

Carry-free addition

- Carry-free → no carry propagation, all digit additions can be done simultaneously
- Carry-free addition is possible with widening of the digit set

1	2	3	4	5	6	radix-10, digit set [0,9]
4	5	6	7	8	9	radix-10, digit set [0,9]
5	7	9	11	13	15	radix-10, digit set [0,18]

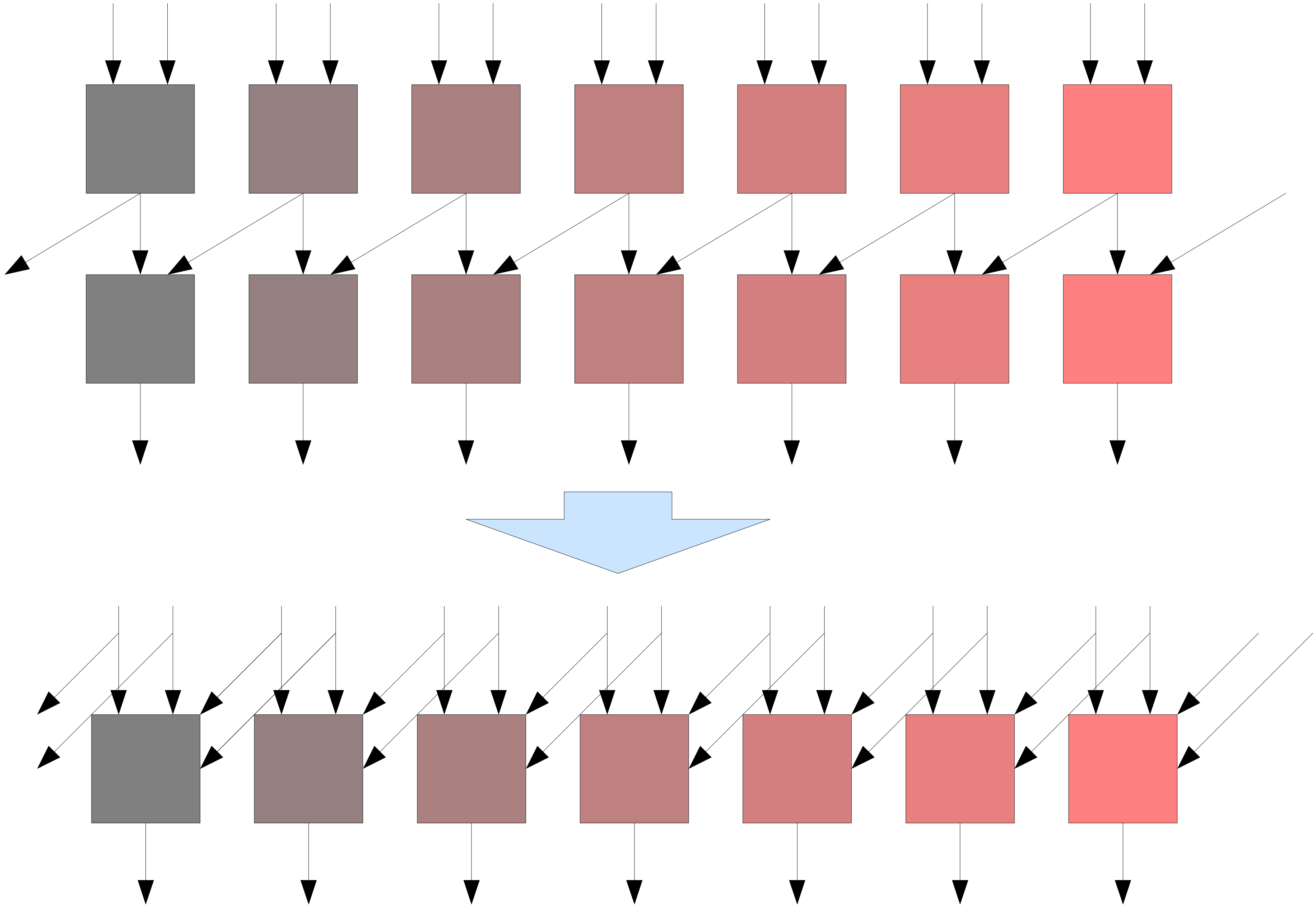


Two-stage carry-free addition

- Reduction of digit set by carry propagation by only one position

11	9	17	10	12	18	radix-10, digit set [0,18]
6	12	9	10	8	18	radix-10, digit set [0,18]
17	21	26	20	20	36	radix-10, digit set [0,36]
7	11	16	0	10	16	Intermediate sums [0,16]
1	1	2	1	2		Transfer digits [0,2]
8	12	18	1	12	16	Sum [0,18]

Two-stage carry-free addition

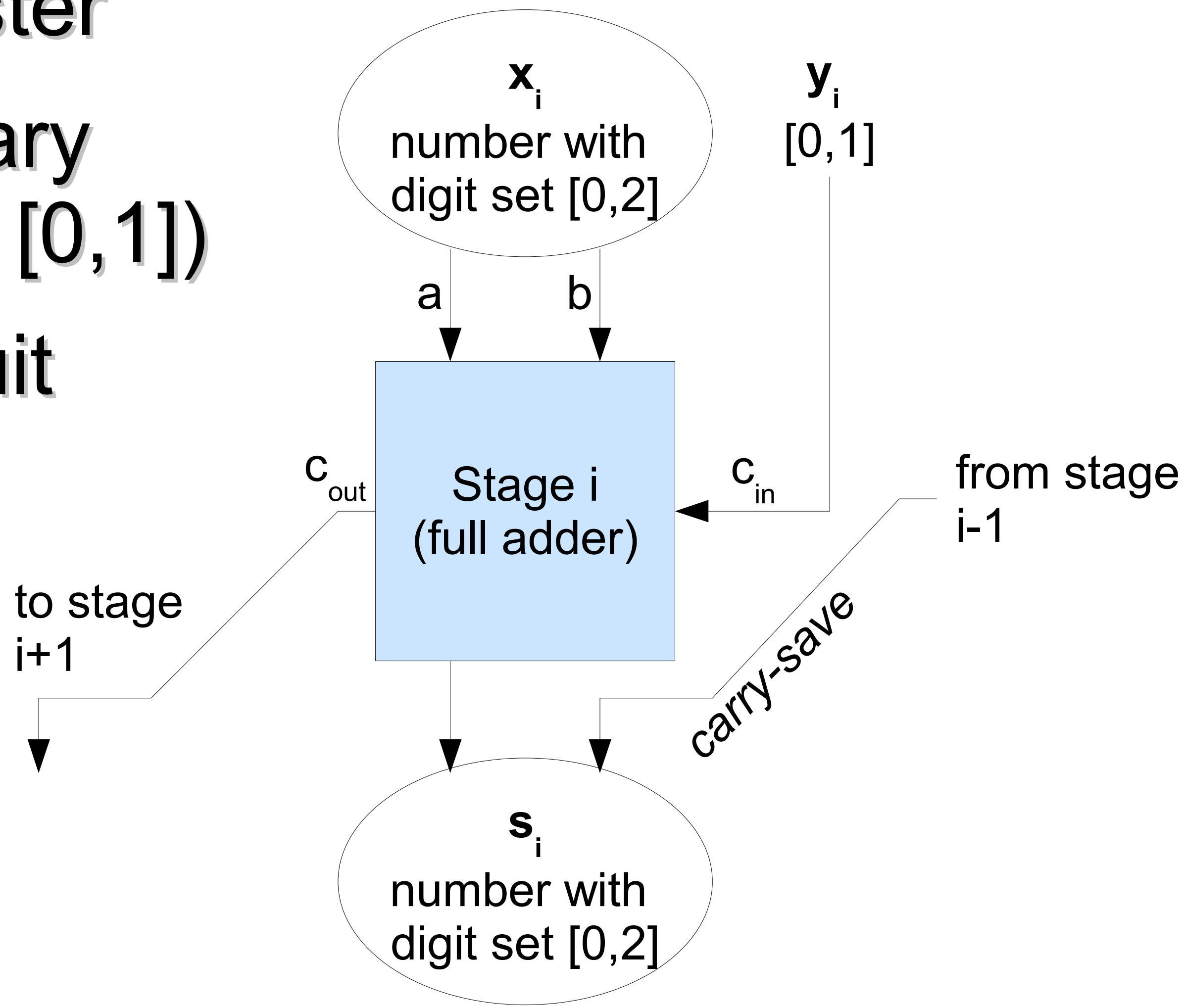


Redundant binary additions

		0	0	1	0	0	1	1^{st} number, radix-2, digit set [0,1]
+		0	1	1	1	1	0	2^{nd} number, radix-2, digit set [0,1]
		0	1	2	1	1	1	Temp. sum, radix-2, digit set [0,2]
+		0	1	1	1	0	1	3^{rd} number, radix-2, digit set [0,1] ([0,2])
		0	2	3	2	1	2	Temp. sum, radix-2, digit set [0,3]
		↓	↓	↓	↓	↓	↓	
		0	0	1	0	1	0	Temp. sums [0,1]
	↙	↙	↙	↙	↙	↙		
	0	1	1	1	0	1	0	Transfer digits [0,1]
		1	1	2	0	2	0	Temp. sum [0,2]
+		0	0	1	0	1	1	4^{th} number, radix-2, digit set [0,1] ([0,2])
		1	1	3	0	3	1	Temp. sum, radix-2, digit set [0,3]
		↓	↓	↓	↓	↓	↓	
		1	1	1	0	1	1	Temp. sums [0,1]
	↙	↙	↙	↙	↙	↙		
	0	0	1	0	1	0		Transfer digits [0,1]
		1	2	1	1	1	1	Result of multiple addition [0,2]
		←	←	←	←	←	←	Carry-propagate conversion
	1	0	0	1	1	1	1	Result of multiple addition [0,1]

Carry-save adder

- Carry is saved in next stage register
- Circuit adds 3 binary numbers (digit set [0,1])
- 3/2 reduction circuit



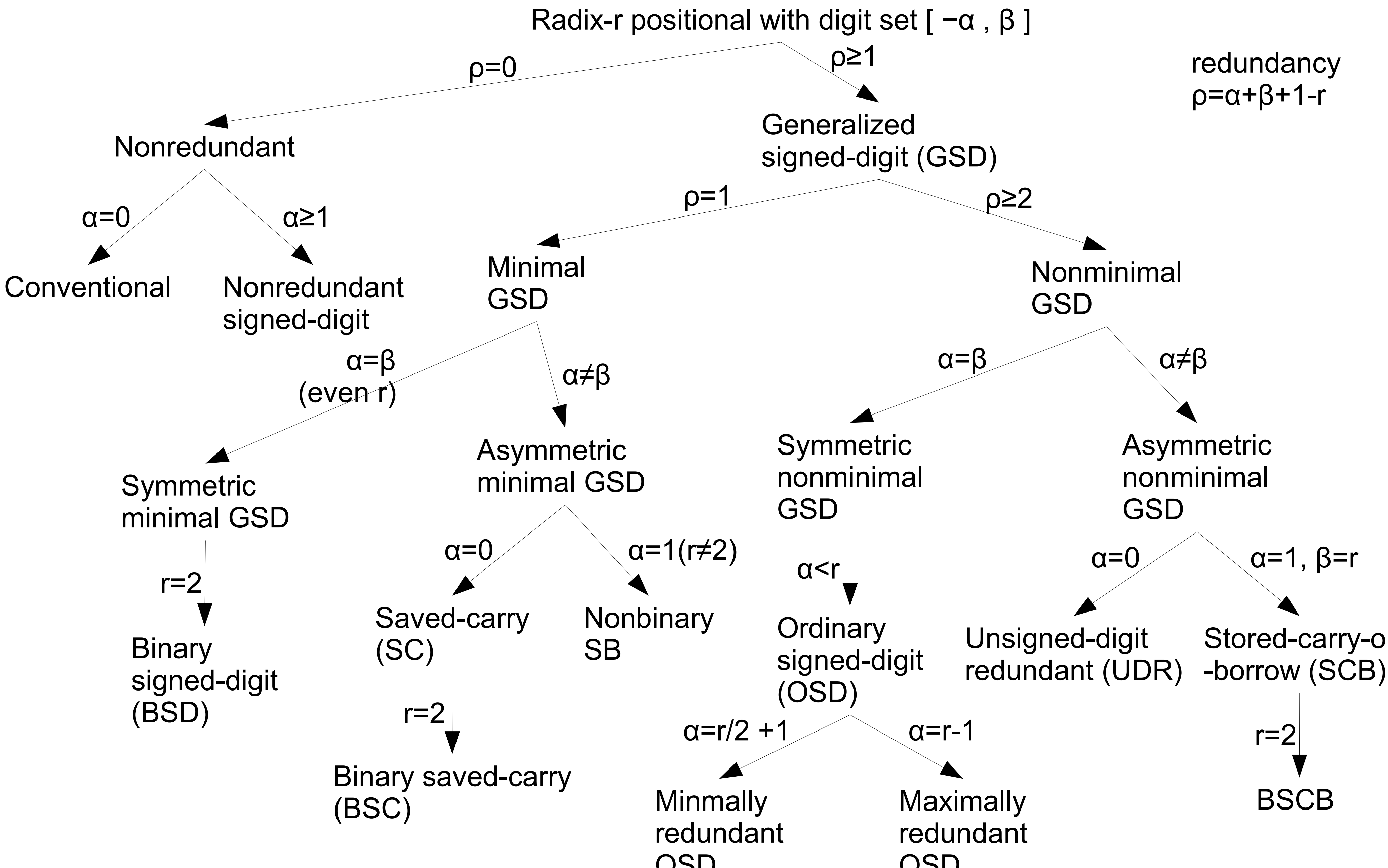
Signed-digit numbers

- All digits have weights r^p (p-position, r-radix)
- Digits can have signed values
- Any set digit [−α,β] including 0, can be used
- If α+β+1>r the numbering system is redundant

```
[-1,1] radix-2 → 1 -1 0 -1 0 = six
[-1,3] radix-4 → 1 -1 2  0 3 = two hundred twenty seven

1111 (2's compl.) → -1 1 1 1 (BSD [-1,1])
```

Number system taxonomy



Implementation of GSD

- Multivalued logic realizations - complex
- Binary encoding schemes for GSD digits
  - sign+value
    - BSD: -1 (11), 0 (00), 1 (01)
  - 2's complement
    - BSD: -1 (11), 0 (00), 1 (01)
  - negative-positive flags ([-1,1] only)
    - BSD: -1 (10), 0 (00), 1 (01)
  - 1-out-of-n
    - BSD: -1 (100), 0 (010), 1 (001)

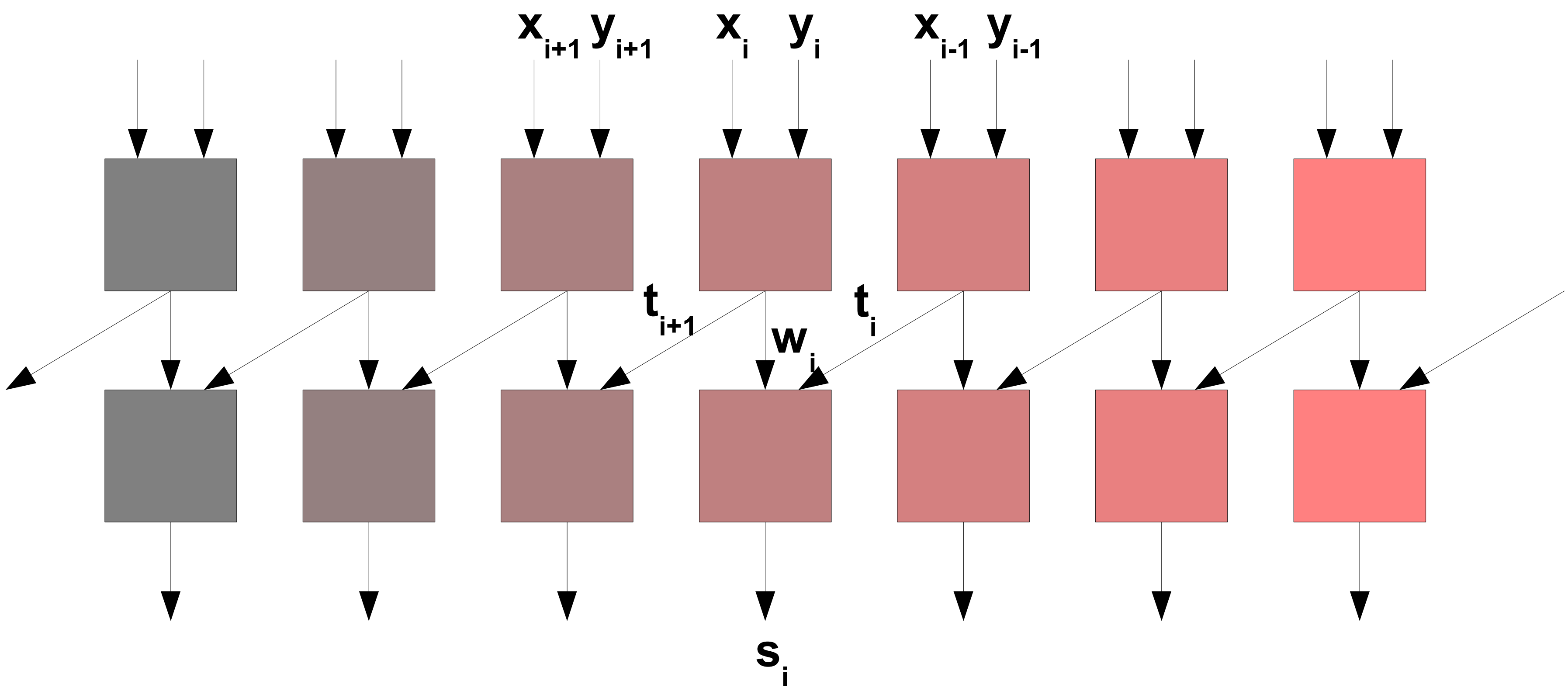
Carry-free addition algorithm

- S = X + Y (GSD numbers with digit set [−α,β])

  p – position sum, w – temp. sum, t – transfer digit, s – result digit

- Compute $p_i = x_i + y_i$
- Compute w_i and t_{i+1}, so that $w_i = p_i - r \cdot t_{i+1}$
  - assume [−λ,μ] digit set for t_{i+1}
- Compute $s_i = w_i + t_i$
  - no new transfer should occur (w_i must absorb t_i)
  - therefore [−(α−λ), β−μ] digit set for w_i
  - $-\alpha+\lambda \le p_i - r \cdot t_{i+1} \le \beta-\mu$ (absorption of value between −λ and μ)
  - $\lambda \ge \alpha/(r-1)$ and $\mu \ge \beta/(r-1)$

Carry-free addition



Carry-free example

- S=X+Y: [-5,9], r=10, ρ=5,
- T: [−λ,μ] = (λ≥5/9,μ≥1) = [-1,1]
- W: [−(α−λ), β−μ] = [-4,8]
- W always absorbs T
- mult. representation of W and T

		3	-4	9	-2	8	X: digit set [-5,9]
+		8	-4	9	8	1	Y: digit set [-5,9]
		11	-8	18	6	9	P: [-10,18]
		1	2	8	6	-1	W: [-4,8]
	1	-1	1	0	1		T: [-1,1]
	1	0	3	8	7	-1	S: [-5,9]

p_i	t_{i+1}	w_i	t_{i+1}	w_i
-10...-5	-1	n		
-4	0	-4	-1	6
-3	0	-3	-1	7
-2	0	-2	-1	8
-1	0	-1		
0	0	0		
1	0	1		
2	0	2		
3	0	3		
4	0	4		
5	0	5		
6	0	6	1	-4
7	0	7	1	-3
8	0	8	1	-2
9...18	1	n		

Carry-free limitations

- Computation of w_i and t_{i+1}, is ambiguous and transfer digits may not be absorbed by w_i if redundancy is low (<3) or radix is small (<3)
- In such a case, an estimation (e_i) of transfer digit from the proceeding position is needed
- Final computation of t_{i+1}, requires information from two previous stages: , e_i and $t_i(e_{i-1})$
- **Limited-carry** addition algorithm for GSD numbers with digit set $[-\alpha,\beta]$

Limited-carry example

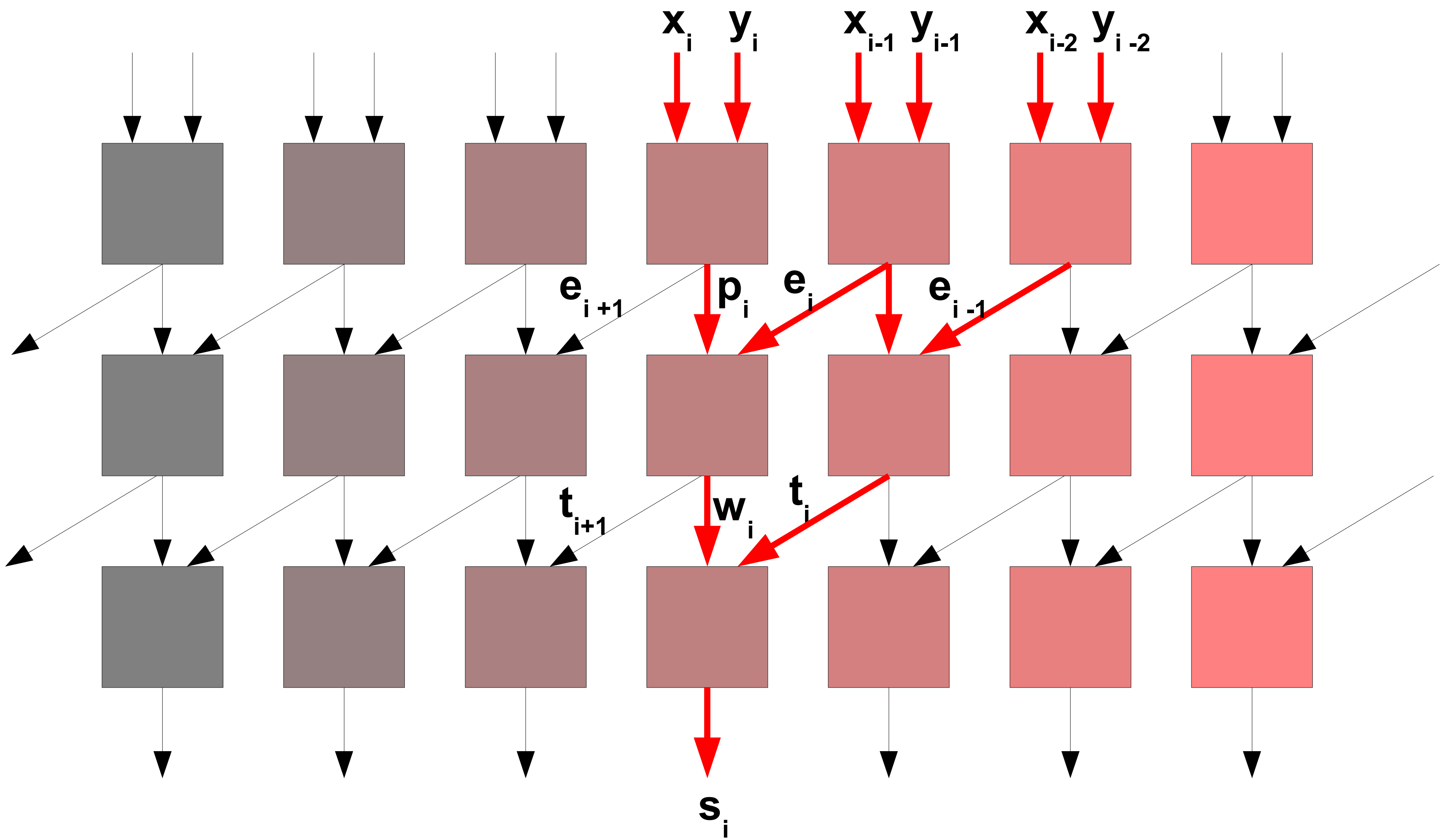
- S=X+Y: [-1,1], r=2, ρ=1,
- T: [−λ,μ] = [-1,1]
- W: [−(α−λ), β−μ] = [0,0] - no absorption-safe digit set !
- W may not always absorb T
- mult. representation of W and T

		1	−1	0	−1	0	X: digit set [-1,1]
+		0	−1	−1	0	1	Y: digit set [-1,1]
		1	−2	−1	−1	1	P: [-2,2]
	high	low	low	low	high		Estimates: low (-2,-1), high(1,2)
		1	0	1	−1	−1	W: [-1,1]
	0	−1	−1	0	1		T: [-1,1]
	0	0	−1	1	0	−1	S: [-1,1]

	high		low	
p_i	t_{i+1}	w_i	t_{i+1}	w_i
−2	−1	0	−1	0
−1	0	−1	−1	1
0	0	0	0	0
1	1	−1	0	1
2	1	0	1	0

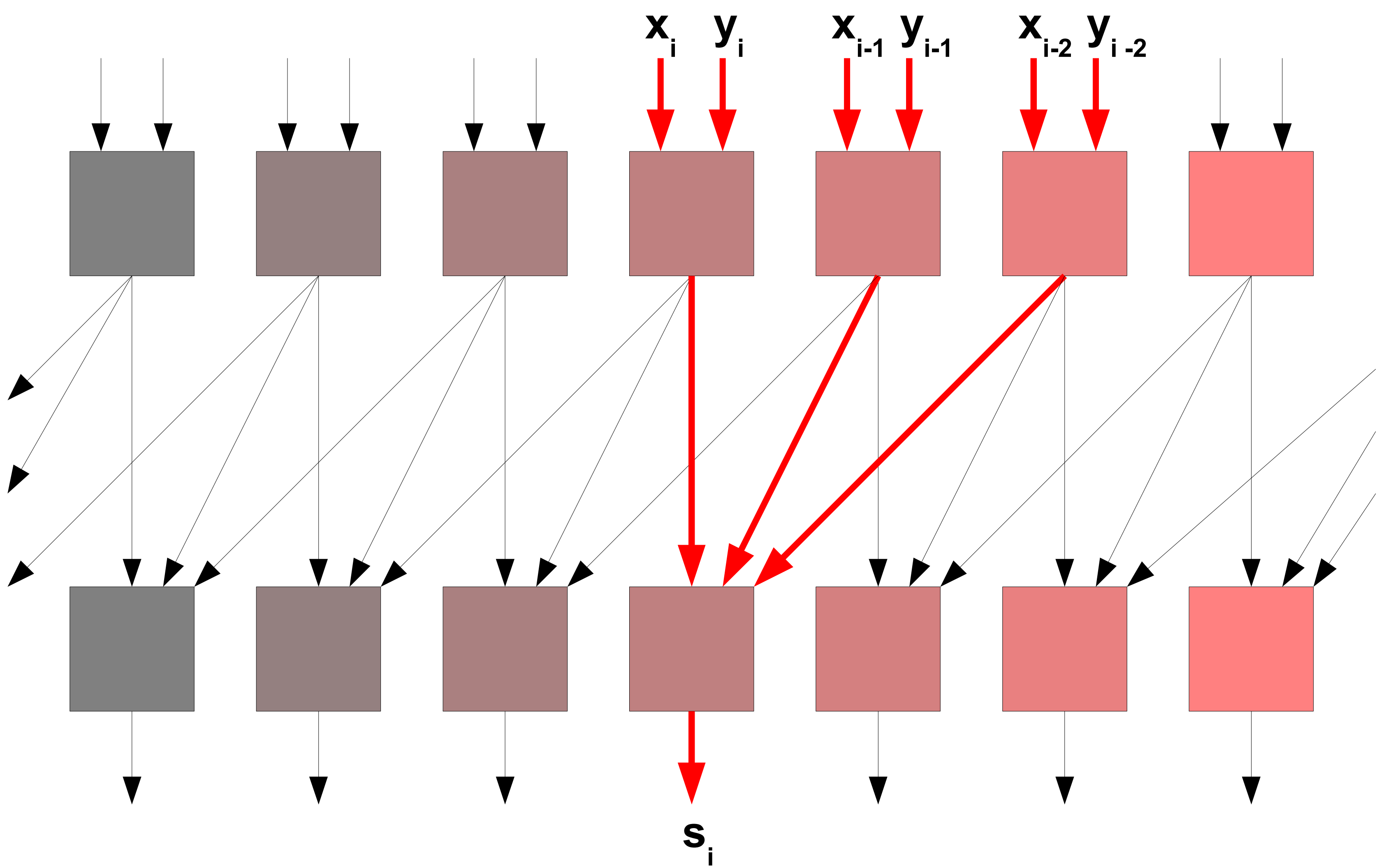
Limited-carry implementation

Three-stage carry-estimate



Limited-carry implementation

Two-stage parallel-carry



Parallel-carry example

- S=X+Y: [0,3], r=2, ρ=2,
- T: [−λ,μ] = [0,3]
- W: [−(α−λ), β−μ] = [0,0] - no absorption-safe digit set !
- all w_i and $t_{i+1,i+2}$ are [0,1], thus the sum is [0,3]

		1	1	3	1	2	X: digit set [0,3]
+		0	0	2	2	1	Y: digit set [0,3]
		1	1	5	3	3	P: [0,6]
		1	1	1	1	1	W: [0,1]
	0	0	0	1	1		T(i+1): [0,1]
0	0	1	0	0			T(i+2): [0,1]
0	0	2	1	2	2	1	S: [0,3]

p_i	t_{i+2}	t_{i+1}	w_i
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0

Subtraction of GSD numbers

- $X-Y = X+(-Y)$
- Negation of GSD numbers can be done by carry-free like algorithm
- Negation of numbers with symmetric digit sets $[-\alpha,\alpha]$ is just inverting the signs of digits

```
BSD [-1,1]

  X    -1   0   1  -1   0
 -X     1   0  -1   1   0
```

X	-1	0	1	-1	0	minus fourteen
-X	1	0	-1	1	0	fourteen