

# Operasi dan Rangkaian Aritmatika

Kuliah#9 TKG205 Sistem Digital - TA 2013/2014

Eko Didik Widianto

Sistem Komputer - Universitas Diponegoro

21 Maret 2014

# Review Kuliah

- ▶ Di kuliah sebelumnya dibahas tentang:
  - ▶ Representasi bilangan biner untuk bilangan bulat dan pecahan, baik tak bertanda maupun bertanda
  - ▶ Bilangan bertanda dinyatakan dalam sign-magnitude, 1's complement dan 2's complement
  - ▶ Bilangan pecahan dinyatakan dengan fixed-point, floating-point 32-bit dan 64-bit
- ▶ Selanjutnya akan dibahas tentang:
  - ▶ Operasi penjumlahan dan pengurangan bilangan biner serta unit penjumlah/pengurang
  - ▶ Rangkaian unit penjumlah/pengurang (ALU, *Arithmetic Logic Unit*)
    - ▶ Penyusun: HA (half-adder), FA (full-adder)
  - ▶ **Susunan rangkaian penjumlah ripple-carry (RCA)**
    - ▶ Rangkaian RCA mempunyai kekurangan terkait delay yang ditimbulkan
  - ▶ Rangkaian **fast adder CLA (*carry-lookahead adder*)** sebagai pengganti RCA

# Review Kuliah: Operasi Bilangan

Operasi dan  
Rangkaian Aritmatika

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

- ▶ Operasi: penjumlahan dan pengurangan
  - ▶ Penjumlahan  $A+B$  menghasilkan hasil jumlah (Sum) dan simpanan (Carry)
  - ▶ Pengurangan  $A-B = A+(-B)$ .  $(-B)$  dinyatakan dengan 2's complement
  - ▶ Rangkaian penjumlah/pengurang  $n$  bit dapat diwujudkan dengan  $n$  buah FA (*full adder*, penjumlah penuh)
  - ▶ Kondisi overflow terjadi jika hasil operasi berada di luar jangkauan bilangan
    - ▶ Perlu rangkaian detektor overflow

# Tentang Kuliah

- ▶ unit penjumlah 1 bit
- ▶ operasi penjumlahan dan pengurangan bilangan biner
- ▶ kondisi *overflow* dalam operasi aritmetika
- ▶ unit penjumlah/pengurang  $n$  bit
- ▶ rangkaian penjumlah/pengurang dengan deteksi *overflow*
- ▶ desain penjumlah cepat  $n$  bit
- ▶ desain dan simulasi penjumlah cepat 32 bit

Operasi dan  
Rangkaian Aritmatika

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

# Kompetensi Dasar

- ▶ Setelah mempelajari bab ini, mahasiswa akan mampu:
  1. [C3] melakukan operasi penjumlahan dan pengurangan bilangan biner
  2. [C5] menganalisis rangkaian penjumlah/pengurang bilangan biner
  3. [C5] mampu menganalisis kondisi overflow dalam suatu operasi aritmetika
  4. [C5] menganalisis rangkaian penjumlah/pengurang n-bit dengan deteksi overflow
  5. [C6] mendesain dan menganalisis rangkaian penjumlah cepat n-bit
- ▶ Link
  - ▶ Website: <http://didik.blog.undip.ac.id/2014/02/25/tkc205-sistem-digital-2013-genap/>
  - ▶ Email: [didik@undip.ac.id](mailto:didik@undip.ac.id)

Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

# Bahasan

Unit Penjumlah 1 Bit

Operasi Aritmetika Biner

Rangkaian Aritmetika Biner

Overflow Aritmatika

Rangkaian Komparator Biner

Rangkaian Penjumlah Cepat

Rangkaian Penjumlah Bilangan Lainnya

Implementasi Rangkaian Aritmetika

Penutup

Lisensi

Operasi dan  
Rangkaian Aritmatika

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

# Unit Penjumlah

- ▶ Komputer mempunyai unit aritmetik dan logika (ALU, *arithmetic logic unit*) untuk menjalankan operasi aritmetika dan logika dari bilangan biner  $n$  bit
- ▶ Operasi aritmetika penjumlahan dan pengurangan dilakukan oleh rangkaian penjumlah/pengurang
- ▶ Rangkaian penjumlah/pengurang ini tersusun atas unit terkecil, yaitu unit penjumlah 1 bit
  - ▶ Unit penjumlah 1 bit dapat diwujudkan dengan rangkaian penjumlah setengah (HA, *half-adder*); dan/atau
  - ▶ rangkaian penjumlah penuh (FA, *full-adder*).

- ▶ Operasi penjumlahan 2 bilangan 1-bit memberikan **4 kombinasi yang mungkin**
  - ▶ Terdapat 2 masukan: **x** dan **y**
  - ▶ Menghasilkan **Sum** dan **Carry-out**
- ▶ **Diimplementasikan** dengan HA (*Half-Adder*)
  - ▶ Disebut HA karena tidak ada **carry-in** yang berasal dari unit penjumlah sebelumnya

Unit Penjumlahan 1 Bit

## Operasi Aritmetika Biner

Rangkaian Aritmetika Biner

## Overflow Aritmatika

## Rangkaian Komparator Biner

### Rangkaian Penjumlahan Cepat

### Rangkaian Penjumlahan Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

## Penutup

Lisensi

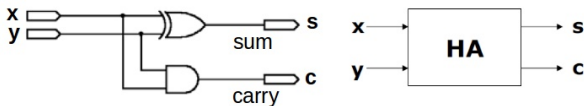
$$\begin{array}{r}
 x \quad 0 \quad 0 \quad 1 \quad 1 \\
 + y \quad + 0 \quad + 1 \quad + 0 \quad + 1 \\
 \hline
 c \ s \quad 0 \ 0 \quad 0 \ 1 \quad 0 \ 1 \quad 1 \ 0
 \end{array}$$

carry sum       $c = x \cdot y$

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



# Rangkaian Half-Adder (HA)



$$\begin{aligned}c &= x \cdot y \\s &= \bar{x} \cdot y + x \cdot \bar{y} \\&= x \oplus y\end{aligned}$$

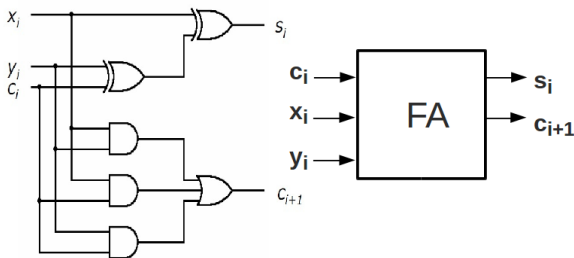
# Operasi Full-Adder (FA)

- Rangkaian FA menyertakan carry-in yang berasal dari carry-out FA sebelumnya
- Terdapat 3 masukan: **x**, **y** dan **carry-in**. Selanjutnya dinotasikan dengan  $x_i$ ,  $y_i$  dan  $c_i$
- Menghasilkan 2 keluaran: **sum** dan **carry-out**. Selanjutnya dinotasikan dengan  $s_i$  dan  $c_{i+1}$

$c_i$	$x_i$	$y_i$	$c_{i+1}$	$s_i$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$c_i$	$x_i$	$y_i$	$c_{i+1}$	$s_i$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

# Rangkaian Full-Adder (FA)



$$\begin{aligned} c_{i+1} &= \sum m(3, 5, 6, 7) \\ &= x_i y_i + c_i x_i + c_i y_i \end{aligned}$$

$$\begin{aligned} s_i &= \sum m(1, 2, 4, 7) \\ &= \bar{c}_i \bar{x}_i y_i + \bar{c}_i x_i \bar{y}_i + c_i \bar{x}_i \bar{y}_i + c_i x_i y_i \\ &= c_i \oplus x_i \oplus y_i \end{aligned}$$

# Rangkaian Dekomposisi FA

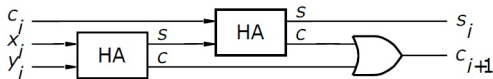


Diagram Blok

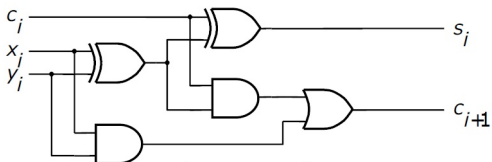


Diagram Detail

$$\begin{aligned}C_{i+1} &= x_i y_i + C_i (x_i \oplus y_i) \\&= x_i y_i + C_i \bar{x}_i y_i + C_i x_i \bar{y}_i \\S_i &= C_i \oplus x_i \oplus y_i\end{aligned}$$

# Fungsi Full-Adder (Dekomposisi Lain)

## ► Persamaan:

$$\text{► } c_{i+1} = x_i y_i + x_i c_i + y_i c_i = x_i y_i + (x_i + y_i) c_i$$

$$\text{► } s_i = x_i \oplus y_i \oplus c_i$$

## ► Anggap $g_i = x_i y_i$ dan $p_i = x_i + y_i$ , maka $c_{i+1} = g_i + p_i c_i$

- Struktur ini akan digunakan untuk **mengurangi delay rangkaian** di rangkaian adder n-bit, terutama propagasi nilai simpan (*Carry*)
- Digunakan di struktur CLA (*carry look-ahead*)

# Operasi Penjumlahan

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

Unit Penjumlahan 1 Bit

## Operasi Aritmetika Biner

Rangkaian Aritmetika Biner

## Overflow Aritmatika

## Rangkaian Komparator Biner

### Rangkaian Penjumlahan Cepat

### Rangkaian Penjumlahan Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

- ▶ Operasi penjumlahan 2 bilangan dengan  $n$ -bit ( $n > 1$ )
  - ▶ Tiap pasang bit ditambahkan
  - ▶ Untuk tiap posisi bit  $i$ , operasi penjumlahannya mungkin melibatkan sebuah **carry-in** dari bit posisi  $(i-1)$

$X = x_3x_2x_1x_0$	0 1 0 1	(5)
$Y = y_3y_2y_1y_0$	0 1 1 0	(6)
	<hr/>	
	0 0 1 1	Sum
	1 0 0	Carry-in
	<hr/>	
$S = s_3s_2s_1s_0$	1 0 1 1	(11)

# Operasi Bilangan Bertanda

## ► Sign-magnitude

- Misalnya: operasi  $5-2=?$  ekuivalen dengan  $5+(-2)=3$ . Bagaimana implementasinya, apakah  $0101+1010$ ?
- Perlu rangkaian logika untuk membandingkan dan mengurangi bilangan

## ► 1's complement

- Misalnya:  $(-5)+(-2)=(-7)$ . Ekuivalen dengan  $1010+1101=(1)0111$ . Carry **1** harus ditambahkan ke  $0111$  agar menghasilkan  $1000$  ( $=-7$ )
- Perlu koreksi untuk mendapatkan hasil yang benar

## ► 2's complement

- penjumlahan selalu benar

# Operasi Sign-Magnitude

- ▶ Hitung operasi pengurangan  $X-Y$  jika  $X=5$  dan  $Y=2$
- ▶ **Solusi.**  $X = 0101$  dan  $Y = 0010$ . Nilai  $(-Y) = -2 = 1010$
- ▶ Operasi  $X+(-Y)$ , harusnya menghasilkan 3 atau 0011

$b_i$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$	
$C_i$	0	0	0	0		
$X_i$		0	1	0	1	(5)
$-Y_i$		1	0	1	0	+ (-2) +
$S_i$		1	1	1	1	+ (-7)

- ▶ Perlu rangkaian logika untuk membandingkan dan mengurangi bilangan



## Operasi 1's Complement

- ▶ Hitung operasi pengurangan  $X-Y$  jika  $X=5$  dan  $Y=2$
- ▶ **Solusi.**  $X = 0101$  dan  $Y = 0010$ . Nilai  $(-Y) = -2 = 1101$
- ▶ Operasi  $X+(-Y)$ , harusnya menghasilkan 3 atau  $0011$

$$\begin{array}{rcccccc}
 b_i & b_4 & b_3 & b_2 & b_1 & b_0 & \\
 C_i & 1 & 1 & 0 & 1 & & \\
 X_i & & 0 & 1 & 0 & 1 & (5) \\
 -Y_i & & 1 & 1 & 0 & 1 & (-2) \\
 S_i & & \hline & 0 & 0 & 1 & 0 & 2
 \end{array}$$

- ▶ Untuk mengoreksi hasil penjumlahan, nilai simpan  $C_4$  yang bernilai 1 perlu ditambahkan di  $S$ , sehingga menghasilkan hasil akhir  $S = 0011$

## ► Penjumlahan

$$\begin{array}{r} (+5) \\ +(+2) \\ \hline (+7) \end{array} \qquad \begin{array}{r} 0\ 1\ 0\ 1 \\ +0\ 0\ 1\ 0 \\ \hline 0\ 1\ 1\ 1 \end{array}$$

$$\begin{array}{r} (+5) \quad 0 \ 1 \ 0 \ 1 \\ + (-2) \quad + 1 \ 1 \ 1 \ 0 \\ \hline (+3) \quad 1 \ 0 \ 0 \ 1 \ 1 \end{array}$$

## Abaikan

$$\begin{array}{r} (-5) \\ + (+2) \\ \hline (-3) \end{array} \qquad \begin{array}{r} 1\ 0\ 1\ 1 \\ +\ 0\ 0\ 1\ 0 \\ \hline 1\ 1\ 0\ 1 \end{array}$$

$$\begin{array}{r} (-5) \\ + (-2) \\ \hline (-7) \end{array} \qquad \begin{array}{r} 1011 \\ + 1110 \\ \hline 11001 \end{array}$$

Abaikan

## ► Pengurangan

$$\begin{array}{r} (+5) \quad 0\ 1\ 0\ 1 \\ -(+2) \quad -0\ 0\ 1\ 0 \\ \hline (+3) \end{array} \Rightarrow \begin{array}{r} 0\ 1\ 0\ 1 \\ +1\ 1\ 1\ 0 \\ \hline 1\ 0\ 0\ 1\ 1 \end{array}$$

Abaikan

$$\begin{array}{r} (-5) \quad 1\ 0\ 1\ 1 \\ -(-2) \quad -1\ 1\ 1\ 0 \\ \hline (-3) \end{array} \Rightarrow \begin{array}{r} 1\ 0\ 1\ 1 \\ +0\ 0\ 1\ 0 \\ \hline 1\ 1\ 0\ 1 \end{array}$$

## Penjumlahan Fixed-Point

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

Unit Penjumlahan 1 Bit

## Operasi Aritmetika Biner

## Rangkaian Aritmetika Biner

## Overflow Aritmatika

## Rangkaian Komparator Biner

## Rangkaian Penjumlahan Cepat

### Rangkaian Penjumlahan Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

- ▶ Diketahui bilangan *fixed-point*  $Q(3,4)$ ,  $A = -1,625$  dan  $B = 38h$ . Tentukan hasil dari operasi berikut:  $A + B$  dan  $A - B$
- ▶ **Solusi.**  $-A = 1,625 = 0001\_1010$ , sehingga  $A = 1110\_0110$   
 Nilai  $B = 0011\_1000$  atau bernilai pecahan  $B = 3,5$ . Nilai  $-B = 1100\_1000$

$b_j$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$	$b_{-1}$	$b_{-2}$	$b_{-3}$	$b_4$		
$C_j$	1	1	1	0	0	0	0	0			
$A_j$		1	1	1	0	0	1	1	0		(-1,625)
$B_j$		0	0	1	1	1	0	0	0	+	3,500
$S_j$		0	0	0	1	1	1	1	0		1,875

- Operasi pengurangan  $A - B = A + (-B)$  :

$b_j$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$	$b_{-1}$	$b_{-2}$	$b_{-3}$	$b_4$		
$C_i$	1	1	0	0	0	0	0	0			
$A_i$		1	1	1	0	0	1	1	0		(-1,625)
$B_i$		1	1	0	0	1	0	0	0	+	(-3,500)
$S_i$		1	0	1	0	1	1	1	0		(-5,125)

## Recall: Operasi Penjumlahan

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

- ▶ Operasi penjumlahan 2 bilangan dengan n-bit ( $n > 1$ )
  - ▶ Tiap pasang bit ditambahkan
  - ▶ Untuk tiap posisi bit i, operasi penjumlahannya mungkin melibatkan sebuah **carry-in** dari bit posisi (i-1)

$X = x_3x_2x_1x_0$	0 1 0 1	(5)
$Y = y_3y_2y_1y_0$	0 1 1 0	(6)
	0 0 1 1	Sum
	1 0 0	Carry-in
$S = s_3s_2s_1s_0$	1 0 1 1	(11)

Unit Penjumlahan 1 Bit

## Operasi Aritmetika Biner

## Rangkaian Aritmetika Biner

## Overflow Aritmatika

## Rangkaian Komparator Biner

### Rangkaian Penjumlahan Cepat

### Rangkaian Penjumlahan Bilangan Lainnya

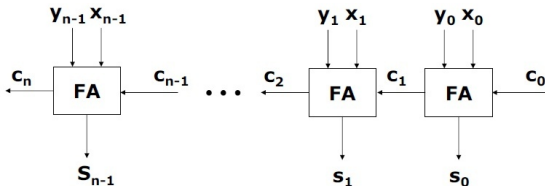
Implementasi  
Rangkaian Aritmetika

## Penutup

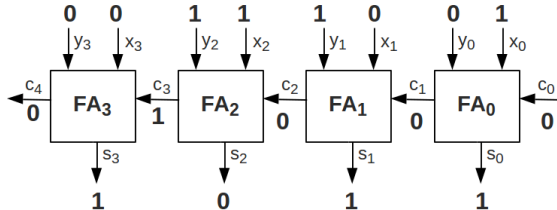
Lisensi

# Rangkaian Ripple Carry Adder (RCA)

- ▶ Operasi penjumlahan dimulai dari pasangan digit paling kanan (LSB) sampai ke paling kiri (MSB)
- ▶ Jika sebuah carry dihasilkan dalam suatu posisi bit  $i$ , maka carry tersebut ditambahkan ke operasi penjumlahan di digit dengan posisi  $i+1$
- ▶ Operasi ini dapat diwujudkan dengan sebuah rantai full-adder (FA) yang dihubungkan seri
  - ▶ Konfigurasi ini disebut sebagai **penjumlah ripple-carry**
  - ▶ Sinyal carry '*ripple*' dari FA satu ke FA berikutnya
  - ▶ RCA ini mempunyai kelemahan, yaitu adanya delay untuk meripple carry
    - ▶ Akan diperbaiki dengan struktur CLA



# Operasi Rangkaian Penjumlah 4 Bit



- ▶ masukan  $X[3 : 0] = 0101$  dan  $Y[3 : 0] = 0110$
- ▶  $FA_0$  mendapatkan masukan  $x_0 = 1$  dan  $y_0 = 0$  serta nilai  $c_0 = 0$  sebagai nilai awalnya
  - ▶ Keluaran dari  $FA_0$  adalah  $s_0 = 1$
  - ▶ membawa simpanan  $c_1 = 0$  ke  $FA_1$
  - ▶ Demikian seterusnya sampai  $FA_3$
- ▶ Rangkaian menghasilkan keluaran  $S[3 : 0] = 1011$  dan  $C[4 : 1] = 0100$

# Unit Penjumlah dan Pengurang

Operasi dan  
Rangkaian Aritmatika

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

- ▶ Operasi pengurangan dapat direalisasikan sebagai operasi penjumlahan dengan menggunakan 2's complement di pengurangnya (baik positif maupun negatif)
  - ▶ Ini memungkinkan menggunakan rangkaian adder untuk **melakukan penjumlahan dan pengurangan sekaligus**
- ▶ Note: (2's complement) = (1's complement) + 1
- ▶ 1's complement dapat diimplementasikan dengan menggunakan XOR
  - ▶  $x \oplus 1 = \bar{x}$  dan  $x \oplus 0 = x$
  - ▶ Jika operasi pengurangan dilakukan, 1's complementkan bilangan kedua dengan meng-XOR-kan semua bit dengan 1



# Unit Penjumlah dan Pengurang

Operasi dan  
Rangkaian Aritmatika

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

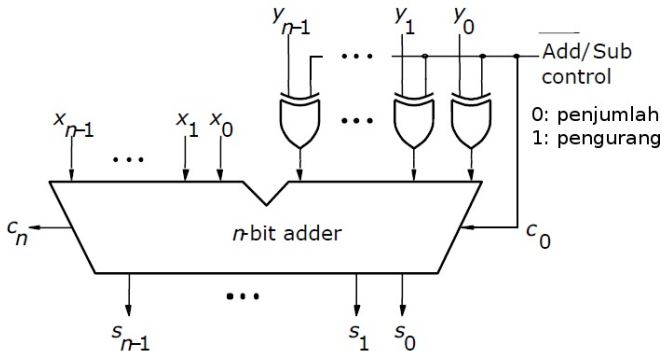
Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

- Menggunakan 2's complement di bilangan pengurang



# Overflow Aritmatika

- ▶ Jika n-bit digunakan untuk merepresentasikan bilangan signed, maka hasil penjumlahan atau pengurangan harus dalam jangkauan  $-2^{n-1}$  sampai  $+2^{n-1} - 1$
- ▶ Jika hasil operasi tidak dalam jangkauan ini, maka telah terjadi **overflow aritmatika**

#Bit, n	Nama	Jangkauan
4	<i>nible, semiocet</i>	<i>signed</i> : $-(2^3)$ s/d $2^3 - 1$ <i>unsigned</i> : 0 s/d $2^4 - 1$
8	<i>byte, octet</i>	<i>signed</i> : $-(2^7)$ s/d $2^7 - 1$ <i>unsigned</i> : 0 s/d $2^8 - 1$
16	<i>half-word, word, short</i>	<i>signed</i> : $-(2^{15})$ s/d $2^{15} - 1$ <i>unsigned</i> : 0 s/d $2^{16} - 1$
32	<i>word, long, doubleword, int</i>	<i>signed</i> : $-(2^{31})$ s/d $2^{31} - 1$ <i>unsigned</i> : 0 s/d $2^{32} - 1$
64	<i>doubleword, int64</i>	<i>signed</i> : $-(2^{63})$ s/d $2^{63} - 1$ <i>unsigned</i> : 0 s/d $2^{64} - 1$
n	Integer n-bit (bentuk umum)	<i>signed</i> : $-(2^{n-1})$ s/d $2^{n-1} - 1$  <i>unsigned</i> : 0 s/d $2^n - 1$

## Overflow Aritmatika

- ▶ Untuk memastikan rangkaian aritmatika beroperasi dengan benar, **perlu pendeteksi** kejadian overflow
  - ▶ Hasil operasi jika overflow, tidak valid

[illegible]

- ▶ Untuk operasi 4-bit, jika c3 dan c4 mempunyai nilai yang sama, maka **tidak terjadi overflow**

# Rangkaian Overflow Aritmatika

Operasi dan  
Rangkaian Aritmatika

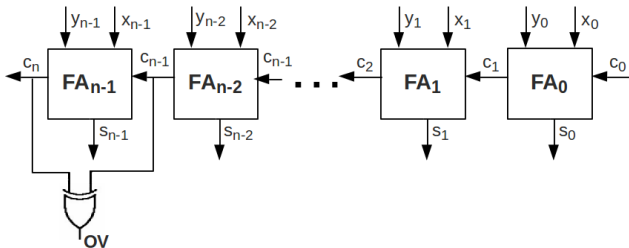
@2014,Eko Didik  
Widianto  
(didik@undip.ac.id)

- ▶ Overflow dapat dideteksi dengan:

- ▶  $\text{overflow} = c_3 \oplus c_4$

- ▶ Untuk bilangan n-bit

- ▶  $\text{overflow} = c_{n-1} \oplus c_n$



Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

- Diketahui bilangan *fixed-point*  $Q(3,4)$ ,  $A = -4,625$  dan  $B = 38h$ .  
Tentukan hasil dari operasi  $A + B$  dan  $A - B$  serta status overflow dari operasi tersebut

$b_j$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$	$b_{-1}$	$b_{-2}$	$b_{-3}$	$b_4$	
$C_j$	0	0	1	1	0	0	0	0		
$A_j$		1	0	1	1	0	1	1	0	(-4,625)
$B_j$		0	0	1	1	1	0	0	0	+ 3,500
$S_j$		1	1	1	0	1	1	1	0	-1,125

$$OV = C_3 \oplus C_4 = 0 \oplus 0 = 0$$

$b_j$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$	$b_{-1}$	$b_{-2}$	$b_{-3}$	$b_4$		
$C_i$	1	0	0	0	0	0	0	0			
$A_i$		1	0	1	1	0	1	1	0	(-4,625)	
$B_i$		1	1	0	0	1	0	0	0	+ (-3,500)	+
$S_i$		0	1	1	1	1	1	1	0	7,875	

$$OV = C_3 \oplus C_4 = 0 \oplus 1 = 1$$

# Rangkaian Komparator Bilangan

Operasi dan  
Rangkaian Aritmatika

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

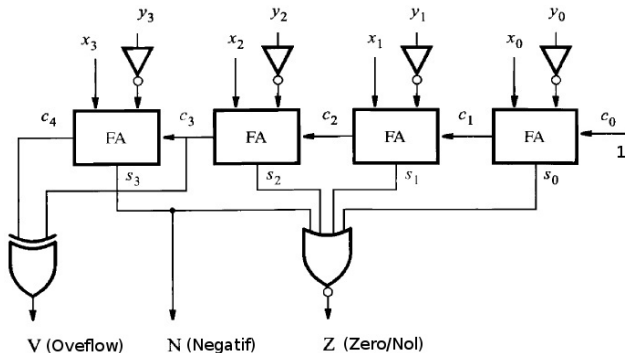
Penutup

Lisensi

- ▶ Misalnya: Bandingkan dua bilangan  $X = x_3x_2x_1x_0$  dan  $Y = y_3y_2y_1y_0$ 
  - ▶ Bisa diimplementasikan dengan rangkaian pengurang (X-Y)
  - ▶ Terdapat 3 output: Z, N, V
    - ▶ Z=1 jika (X-Y)=0, hasil lainnya Z=0
    - ▶ N=1 jika (X-Y)<0, hasil lainnya N=0
    - ▶ V=1 jika terjadi overflow aritmatika, kalau tidak ada overflow V=0
  - ▶ Tunjukkan bagaimana Z, N, V digunakan untuk menentukan  $X=Y$ ,  $X<Y$ ,  $X\leq Y$ ,  $X>Y$  dan  $X\geq Y$

- ▶ Misalnya  $X < Y$ , kemungkinan yang akan terjadi
  - ▶ Jika  $X$  dan  $Y$  mempunyai tanda yang sama, tidak akan terjadi overflow, sehingga  $V=0$ . Dan untuk semua nilai  $X$  dan  $Y$  (positif/negatif) menghasilkan nilai negatif  $N=1$
  - ▶ Saat  $X$  negatif dan  $Y$  positif, maka  $(X-Y)$  akan negatif ( $N=1$ ) jika tidak ada overflow ( $V=0$ ) dan  $(X-Y)$  akan positif ( $N=0$ ) jika terdapat overflow ( $V=1$ )
  - ▶ Sehingga jika  $X < Y$  maka  $N \oplus V = 1$
- ▶ Hasil
  - ▶  $X=Y$  terdeteksi saat  $Z=1$
  - ▶  $X < Y$  terdeteksi jika  $N \oplus V = 1$ .  $X \leq Y$  terdeteksi jika  $Z + (N \oplus V) = 1$
  - ▶  $X > Y$  terdeteksi jika  $\overline{Z + (N \oplus V)} = 1$ .  $X \geq Y$  terdeteksi jika  $\overline{(N \oplus V)} = 1$

# Rangkaian Komparator 4-bit



Operasi dan  
Rangkaian Aritmatika

@2014,Eko Didik  
Widianto  
(didik@undip.ac.id)

Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmatika

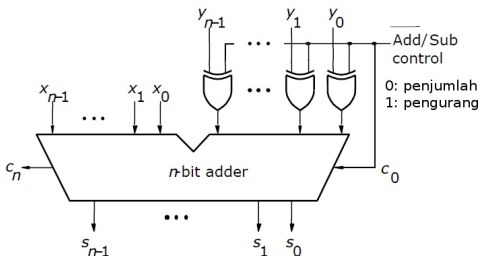
Penutup

Lisensi



# Isu Performansi

- ▶ Penjumlahan dan pengurangan merupakan **operasi dasar di sistem komputer** sebagai perangkat komputasi
  - ▶ Performansi operasi ini (mis: kecepatan) membawa pengaruh signifikan terhadap performansi keseluruhan
  - ▶ Meningkatkan performansi dapat menggunakan **rangkaian yang lebih cepat**
    - ▶ Menggunakan teknologi terbaru yang mengurangi delay gerbang dasar
  - ▶ Performansi bisa diperoleh dengan mengubah **struktur rangkaian fungsional**



# Kekurangan Ripple Carry Adder

- ▶ Tiap FA mempunyai delay tertentu sebelum keluaran  $s_i$  dan  $c_{i+1}$  valid
  - ▶ disebut **delay propagasi FA** dari input ke output
- ▶ Misalnya, delay propagasi  $\Delta t$ 
  - ▶ Carry dari FA pertama,  $c_1$ , akan sampai di FA kedua dalam waktu  $\Delta t$  setelah input  $x_0$  dan  $y_0$
  - ▶ Carry dari FA kedua,  $c_2$ , akan sampai di FA ketiga dalam waktu  $\Delta t$  setelah input  $x_1$ ,  $y_1$  dan  $c_1$  atau total  $2\Delta t$
  - ▶ Dan seterusnya. Sinyal  $c_{n-1}$  valid setelah  $(n-1)\Delta t$ , dan jumlah total akan tersedia setelah delay  $(n)\Delta t$
- ▶ Delay total tersebut **semakin besar seiring semakin banyak jumlah bit bilangan** yang harus ditambahkan
  - ▶ Penjumlahan bilangan  $n$ -bit akan membutuhkan waktu  $(n)\Delta t$  dari bit-bit masukan tersedia sampai keluaran valid
- ▶ Delay terbesar dalam rangkaian disebut **critical-path delay**
  - ▶ Jalur yang menyebabkan delay ini disebut **critical path**

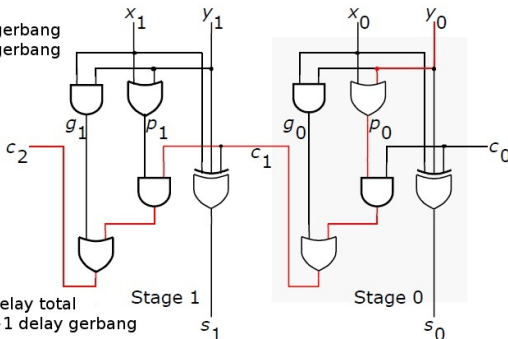
# Performansi Adder/Subtractor

Operasi dan  
Rangkaian Aritmatika

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

- Identifikasi jalur yang menyebabkan delay terbesar (critical path)
- Recall critical path di RCA:

C1: delay 3 gerbang  
C2: delay 5 gerbang  
dst....



RCA n-bit: Delay total  
sebesar  $2n+1$  delay gerbang

Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

Rangkaian Penjumlah  
Bilangan Lainnya

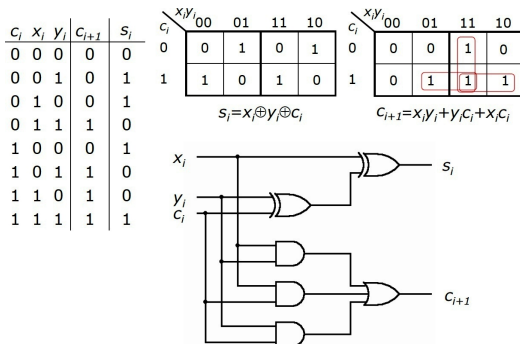
Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

# Carry-lookahead Adder/CLA

- Untuk mengurangi delay akibat **propagasi carry** di RCA (*critical-path-delay*)
  - **Evaluasi tiap stage FA apakah carry-in** dari stage sebelumnya akan mempunyai nilai 0 atau 1
  - Jika evaluasi dapat dilakukan dengan cepat, performansi adder dapat ditingkatkan
- Recall FA yang ada di tiap stage:



# Carry-lookahead Adder (CLA)

- ▶ Fungsi carry-out dari stage  $i$  (satu FA) adalah
$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i = x_i y_i + (x_i + y_i) c_i$$
- ▶ Anggap  $g_i = x_i y_i$  dan  $p_i = x_i + y_i$ , maka  $c_{i+1} = g_i + p_i c_i$ 
  - ▶ Fungsi  $g_i = 1$  jika  $x_i = 1$  dan  $y_i = 1$ , tanpa pengaruh  $c_i$ . Stage  $i$  pasti membangkitkan carry-out, sehingga  $g$  disebut fungsi **generate**
  - ▶ Fungsi  $p_i = 1$  jika salah satu  $x_i = 1$  atau  $y_i = 1$  atau keduanya 1. Stage  $i$  membangkitkan carry-out jika  $c_i = 1$ . Nilai  $c_i = 1$  ini dipropagasikan lewat FA di stage  $i$ , sehingga  $p$  disebut fungsi **propagate**

# Carry-lookahead Adder

- ▶ Ekspansi persamaan  $c_{i+1} = g_i + p_i c_i$ . Dengan  $c_i = g_{i-1} + p_{i-1} c_{i-1}$ , akan menghasilkan

$$\begin{aligned}c_{i+1} &= g_i + p_i (g_{i-1} + p_{i-1} c_{i-1}) \\ &= g_i + p_i g_{i-1} + p_i p_{i-1} c_{i-1}\end{aligned}$$

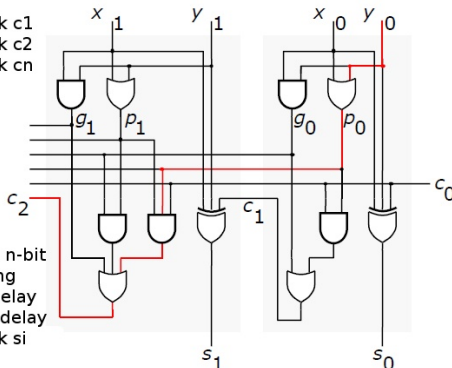
- ▶ Ekspansi sampai stage 0:

$$\begin{aligned}c_{i+1} &= g_i + p_i g_{i-1} + p_i p_{i-1} g_{i-2} + \cdots + p_i p_{i-1} \cdots p_2 p_1 g_0 \\ &\quad + p_i p_{i-1} \cdots p_2 p_1 p_0 c_{i-1}\end{aligned}$$

- ▶ Ekspresi tersebut menggambarkan rangkaian AND-OR 2-level yang memungkinkan  $c_{i+1}$  dapat dihasilkan dengan cepat
  - ▶ Ini disebut **carry-lookahead adder**

# Critical Path CLA

3 delay gerbang untuk  $c_1$   
3 delay gerbang untuk  $c_2$   
3 delay gerbang untuk  $c_n$



Delay total untuk CLA n-bit  
adalah 4 delay gerbang  
- Semua  $g_i$  dan  $p_i$ , 1 delay  
- Semua  $c_i$ , tambah 2 delay  
- Satu delay lagi untuk  $s_i$

# Keterbatasan CLA

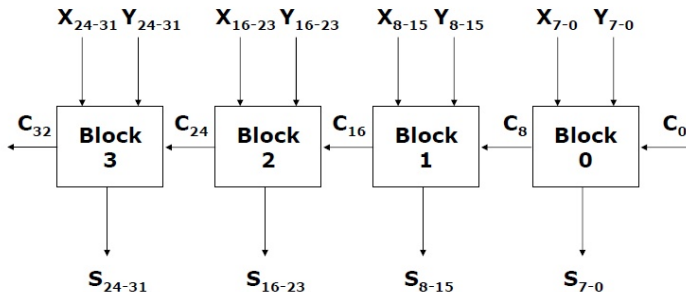
- ▶ Persamaan carry-out di CLA menghasilkan solusi adder yang cepat karena hanya merupakan fungsi AND-OR 2-level
- ▶ Namun, **batasan fan-in** dapat membatasi kecepatan CLA
  - ▶  $FA_0$ : AND dan OR 2-input,  $c_1 = g_0 + p_0 c_0$
  - ▶  $FA_1$ : AND dan OR 3-input,  $c_2 = g_1 + p_1 g_0 + p_1 p_0 c_0$
  - ▶  $FA_2$ : AND dan OR 4-input,  
 $c_3 = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_0$
  - ▶  $FA_n$ : AND dan OR (n+2)-input
- ▶ Device seperti FPGA seringkali menggunakan rangkaian khusus untuk implementasi fast adder
- ▶ **Kompleksitas CLA n-bit akan bertambah jika n bertambah**
  - ▶ Untuk menguranginya, digunakan **pendekatan hirarki** untuk mendesain adder yang lebih besar



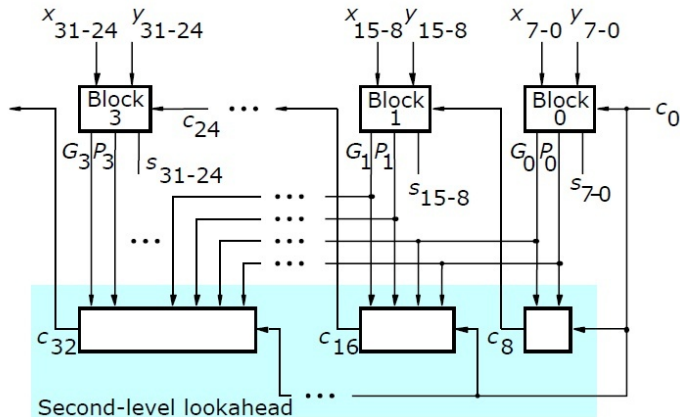
# Desain Adder 32-bit

- ▶ Misalnya diinginkan rangkaian penjumlah 32-bit
- ▶ Bagi adder ini menjadi 4 blok sehingga
  - ▶ Blok 0 untuk operasi bit  $b_7 - b_0$
  - ▶ Blok 1 untuk operasi bit  $b_{15} - b_8$
  - ▶ Blok 2 untuk operasi bit  $b_{23} - b_{16}$
  - ▶ Blok 3 untuk operasi bit  $b_{31} - b_{24}$
- ▶ Tiap blok dibangun dengan adder CLA 8-bit
  - ▶ Carry-out untuk tiap blok adalah  $c_8, c_{16}, c_{24}$  dan  $c_{32}$
- ▶ Terdapat 2 pendekatan untuk menghubungkan ke-empat blok
  - ▶ Ripple-carry
  - ▶ Carry-lookahead level-2

# Ripple-Carry Antar Blok



# Carry-lookahead Level-2



# Carry-lookahead Level-2

## ► Persamaan CLA level-2

$$P_0 = p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0$$

$$G_0 = g_7 + p_7 g_6 + p_7 p_6 g_5 + \cdots + p_7 p_6 p_5 p_4 p_3 p_2 p_1 g_0$$

$$c_8 = G_0 + P_0 c_0$$

$$c_{16} = G_1 + P_1 c_8 = G_1 + P_1 G_0 + P_1 P_0 c_0$$

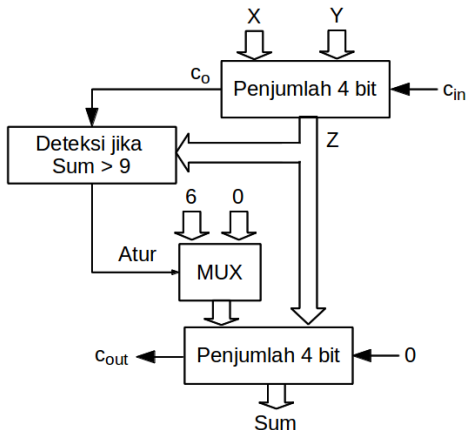
$$c_{24} = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 c_0$$

$$c_{32} = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 c_0$$

# Analisis Rangkaian CLA Hirarki

- ▶ Asumsi konstrain fan-in adalah 4 masukan, waktu yang diperlukan untuk melakukan operasi penambahan 2 bilangan 32-bit meliputi:
  - ▶ Lima delay gerbang untuk membentuk term  $G_i$  dan  $P_i$ , 3 delay gerbang untuk CLA level-2, dan satu delay untuk menghasilkan bit sum akhir
    - ▶ Sebenarnya bit sum final diperoleh setelah 8 delay karena  $c_{32}$  tidak digunakan untuk menghitung bit sum
  - ▶ Operasi lengkap, termasuk deteksi overflow ( $c_{31} \oplus c_{32}$ ), membutuhkan 9 delay gerbang
    - ▶ Bandingkan 65 delay di ripple-carry adder

# Rangkaian Penjumlah BCD



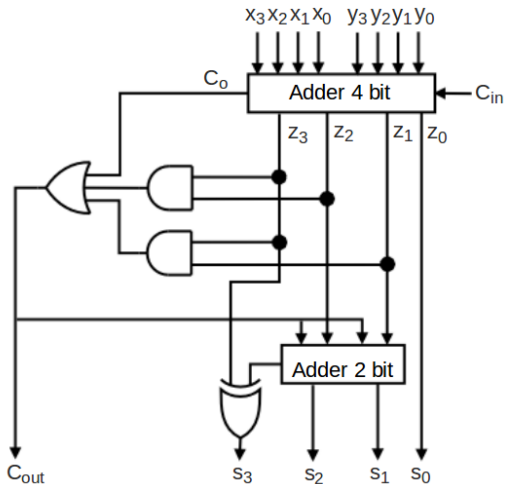
# Operasi Penjumlahan BCD

- ▶ Diketahui bilangan BCD  $X = 6$  dan  $Y = 5$ . Nyatakan hasil penjumlahan BCD  $X + Y$ .
- ▶ **Solusi.** Bilangan BCD  $X = 0110$  dan  $Y = 0101$ . Operasi penjumlahannya adalah sebagai berikut:

$$\begin{array}{r} X \quad \quad 0 \quad 1 \quad 1 \quad 0 \quad \quad 6 \\ Y \quad + \quad 0 \quad 1 \quad 0 \quad 1 \quad + \quad 5 \quad + \\ \hline Z \quad \quad 1 \quad 0 \quad 1 \quad 1 \quad \quad 11 \\ \quad \quad 0 \quad 1 \quad 1 \quad 0 \quad + \\ S \quad 1 \quad \hline 0 \quad 0 \quad 0 \quad 1 \end{array}$$

- ▶ Hasil penjumlahan lebih dari 9, sehingga perlu ditambahkan dengan 6 atau 0110 untuk menghasilkan nilai jumlah  $S = 10001$  yang merupakan bilangan BCD dari desimal 11.

# Rangkaian Penjumlah BCD 1 Digit





# IC TTL untuk Operasi Aritmetika

Operasi dan  
Rangkaian Aritmatika

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

Lisensi

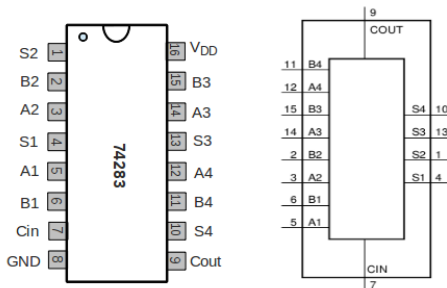
IC TTL	Deskripsi	Contoh
7480	Penjumlah penuh (FA) 1 bit	SN7480 (Texas Instruments)
7482	Penjumlah penuh (FA) 2 bit	NTE7482 (NTE Electronics)
7483/74283	Penjumlah penuh (FA) 4 bit (CLA)	74HC283 (Philips)
74385	Quad penjumlah/pengurang serial 4-bit	74LS385 (Texas Instruments)

# IC 74283

Operasi dan  
Rangkaian Aritmatika

@2014,Eko Didik  
Widianto  
([didik@undip.ac.id](mailto:didik@undip.ac.id))

## ► Struktur CLA, 4 bit



Unit Penjumlah 1 Bit

Operasi Aritmetika  
Biner

Rangkaian Aritmetika  
Biner

Overflow Aritmatika

Rangkaian Komparator  
Biner

Rangkaian Penjumlah  
Cepat

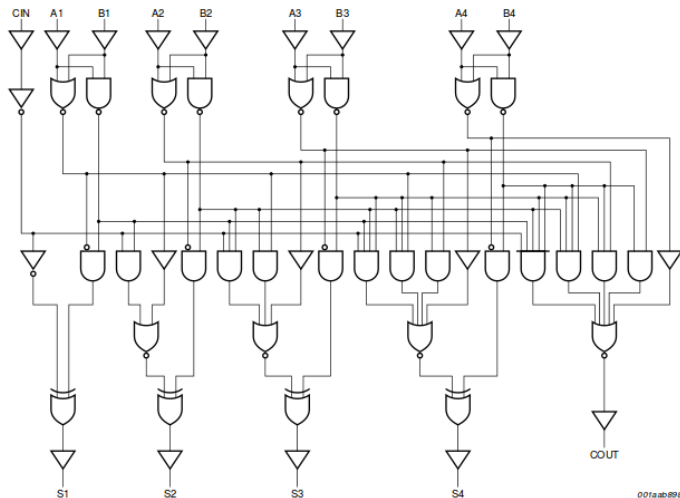
Rangkaian Penjumlah  
Bilangan Lainnya

Implementasi  
Rangkaian Aritmetika

Penutup

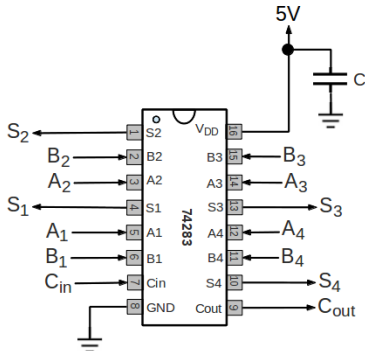
Lisensi

# Struktur IC 74283



Sumber: datasheet Philips 74HC283

# Implementasi Rangkaian Penjumlah 4 Bit



- Implementasikan rangkaian penjumlah/pengurang 4 bit menggunakan 74HC283. Gunakan masukan kontrol Add/Sub untuk mengeset operasi rangkaian sebagai penjumlah atau pengurang
  - (Petunjuk: -B diperoleh dengan menggunakan 4 gerbang XOR-2 dan mengeset  $C_{in} = 1$ )

# Ringkasan Kuliah

- ▶ Yang telah kita pelajari hari ini:
  - ▶ Operasi penjumlahan bilangan biner 1 bit dengan HA dan FA
  - ▶ Operasi penjumlahan/pengurangan bilangan tak bertanda dan bertanda
  - ▶ Rangkaian penjumlah n bit dengan RCA
  - ▶ Overflow dan rangkaian detektornya
  - ▶ Rangkaian penjumlah cepat dengan CLA dan performansinya
  - ▶ Rangkaian penjumlah BCD
  - ▶ Implementasi rangkaian penjumlah dengan IC 7400
- ▶ Bab berikutnya akan operasi aritmetika dan rangkaiannya, meliputi penjumlahan dan pengurangan, menggunakan format bilangan yang telah dijabarkan di bab ini
- ▶ Pelajari: <http://didik.blog.undip.ac.id/2014/02/25/tkc205-sistem-digital-2013-genap/>

## Creative Common Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

- ▶ Anda bebas:
  - ▶ untuk **Membagikan** — untuk menyalin, mendistribusikan, dan menyebarkan karya, dan
  - ▶ untuk **Remix** — untuk mengadaptasikan karya
- ▶ Di bawah persyaratan berikut:
  - ▶ **Atribusi** — Anda harus memberikan atribusi karya sesuai dengan cara-cara yang diminta oleh pembuat karya tersebut atau pihak yang mengeluarkan lisensi. Atribusi yang dimaksud adalah mencantumkan alamat URL di bawah sebagai sumber.
  - ▶ **Pembagian Serupa** — Jika Anda mengubah, menambah, atau membuat karya lain menggunakan karya ini, Anda hanya boleh menyebarkan karya tersebut hanya dengan lisensi yang sama, serupa, atau kompatibel.
- ▶ Lihat: [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/)
- ▶ Alamat URL: <http://didik.blog.undip.ac.id/2014/02/25/tkc205-sistem-digital-2013-genap/>