

# 加算機なのに減算?

ComputerScience集会#14 @VRChat 2023-12-04

夜鍋ヨナ-yonabeyona <<http://x.com/yonabeyona>>

# CS 集会所

# # 14

# 加算器なのに

# 減算？

# 自己紹介

- 名前：夜鍋 ヨナ(よなべ よな)
- X(Twitter) : yonabeyona, yonabeyona\_sub
- Discord : yona\_47
- その他
  - ComputerScienceが好き
  - 数学勉強中
  - 物理も勉強中
  - ComputerScienceの中でも、ComputerArchitectureが好き



# 今日やること

- 復習1 : 全加算器
- 復習2 : 二進数での減算
- 補数変換回路
- 補数変換回路と全加算器の類似性について
- 切り替える回路は存在するか?
- ALUが(半分)完成!
- 次回やることはシフト回路

# 今日やること

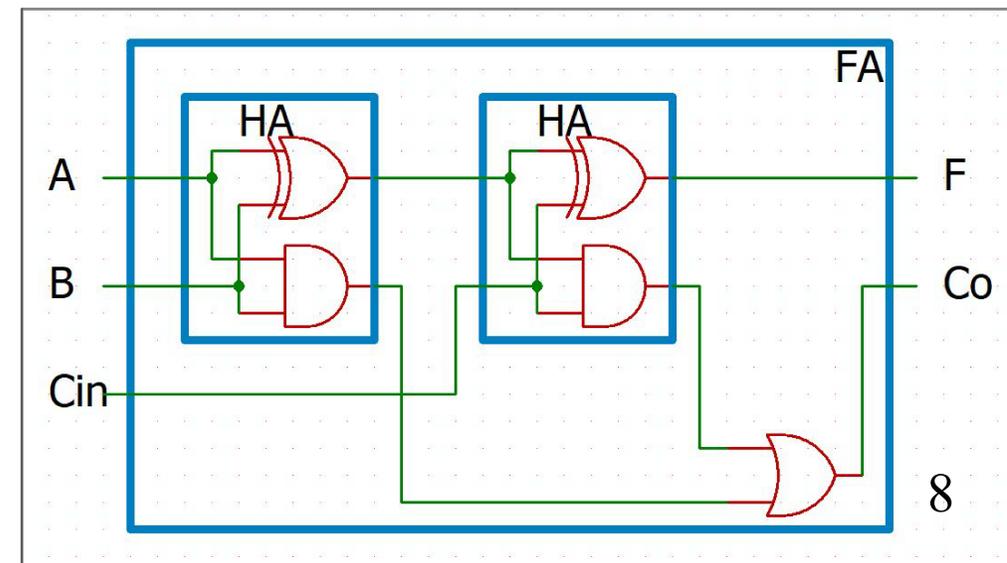
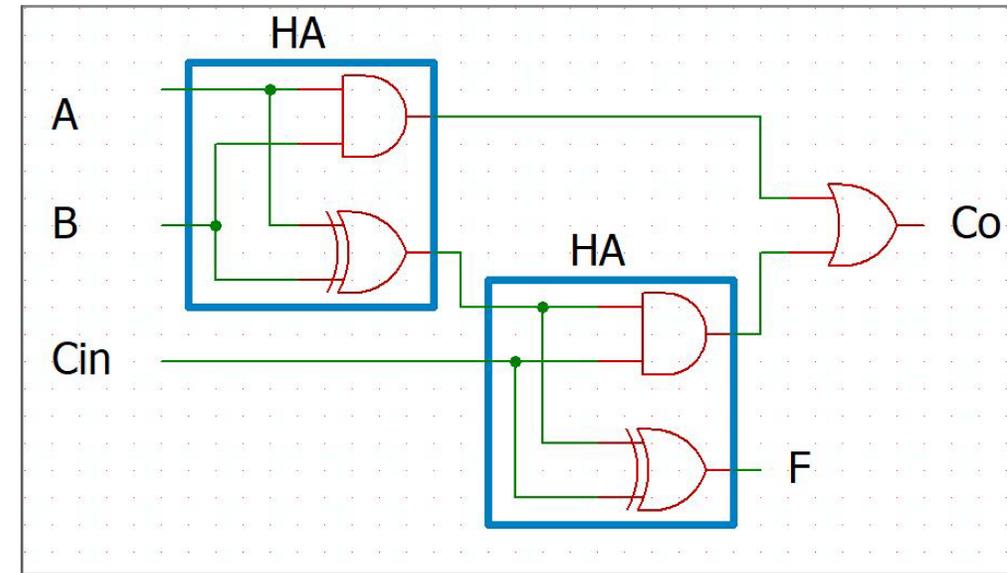
- **復習1 : 全加算器**
- 復習2 : 二進数での減算
- 補数変換回路
- 補数変換回路と全加算器の類似性について
- 切り替える回路は存在するか?
- ALUが(半分)完成!
- 次回やることはシフト回路

# 復習1：全加算器

- 半 + 半 = 全
- Nbit 全加算器

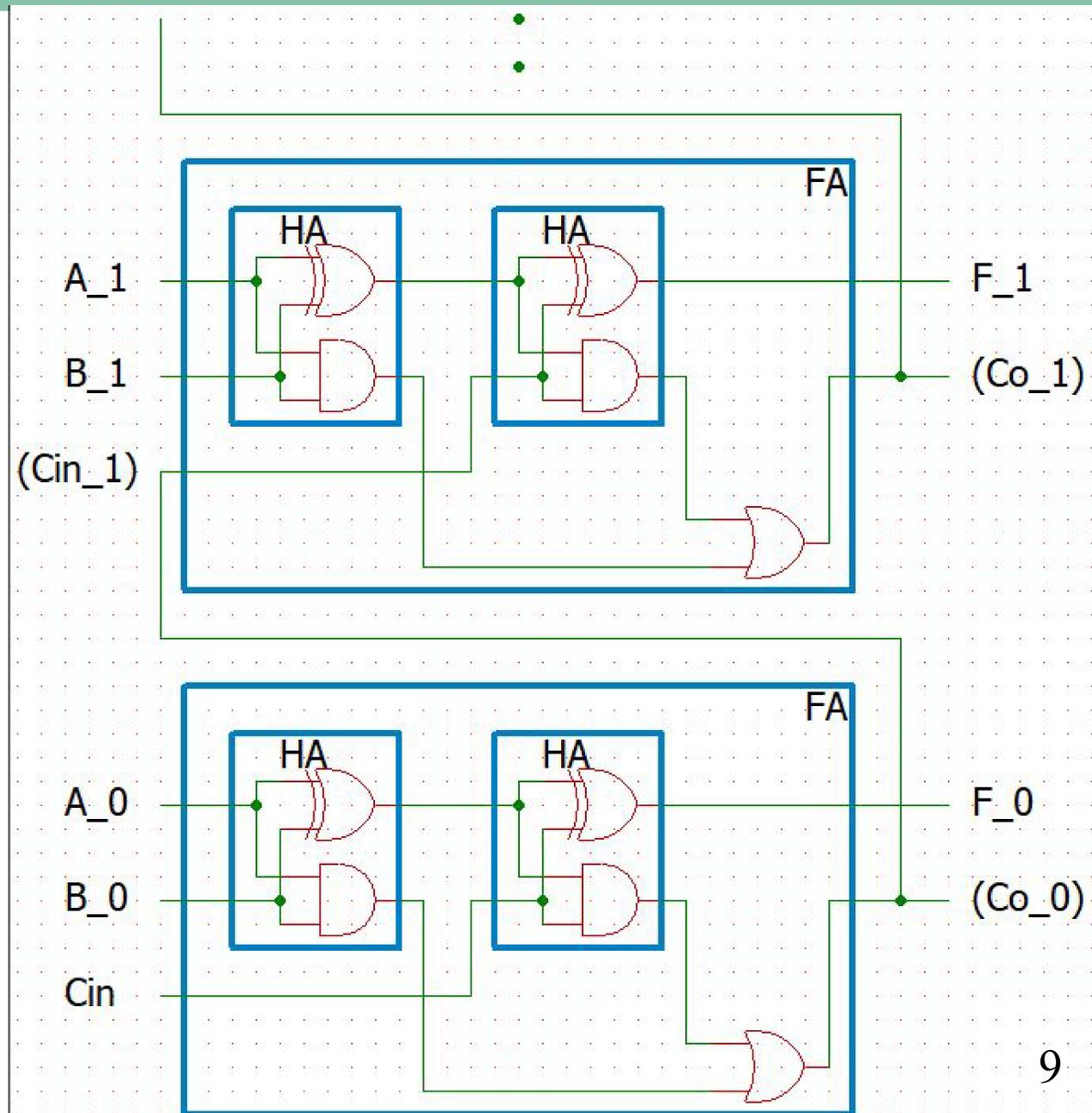
# 半 + 半 = 全

- 全加算器は半加算器を2個使うとできる
- (OR回路も必要)
- これで1bit加算器ができた
- Nbit加算器は?



# Nbit全加算器

- Nbit加算器の作り方
- nビット目のCoを  
n+1ビット目のCinへ接続
- 入力AとBを1ビットずつ  
全加算器に入力
- 最後のCoがいわゆるCフラグ  
オーバーフローとか検知する



# 今日やること

- 復習1 : 全加算器
- **復習2 : 二進数での減算**
- 補数変換回路
- 補数変換回路と全加算器の類似性について
- 切り替える回路は存在するか?
- ALUが(半分)完成!
- 次回やることはシフト回路

# 復習2：2進数での減算

- 補数を使う
- 引き算  $\Leftrightarrow$  2の補数を足す
- 1の補数 と 2の補数

# 補数を使う

補数の定義

言い換えると?

# 補数を使う

## 補数の定義

ある数体系  $N$  における  $n$  桁の数  $B$  について、  
 $N^n - B$  を  $N$  に対する補数、あるいは単に  $B$  の  $N$  の補数という。

言い換えると?

# 補数を使う

## 補数の定義

ある数体系  $N$  における  $n$  桁の数  $B$  について、  
 $N^n - B$  を  $N$  に対する補数、あるいは単に  $B$  の  $N$  の補数という。

## 言い換えると?

補数にしたい  $N$  進数  $x$  桁の数字に「ある数」を足すと、  
(それらよりもう1桁大きい)  $N^x$  な数になるような数を指してるともいえる。

# 負数を足す $\iff$ 引き算

## 2の補数を用いた減算

2の補数を用いた2進数の減算  $A - B$  は、以下の式で表される。

$$\begin{aligned} A - B &= A + (-B) \\ &= A + (-B) + 0 \\ &= A + (-B) + 2^n - 2^n \\ &= A + (2^n - B) - 2^n \end{aligned}$$

hint :  $(2^n - B)$  で補数を求めて、 $-2^n$  で符号の調整をしている。

# 1の補数 と 2の補数

## 1の補数

見た目：あるN桁の数Xに足すと、N桁の「 $111\dots111$ 」になるような数Bのこと

## 2の補数

見た目：あるN桁数Xに足すと、(N+1)桁の「 $100\dots000$ 」になるような数Bのこと

# 今日やること

- 復習1 : 全加算器
- 復習2 : 二進数での減算
- **補数変換回路**
- 補数変換回路と全加算器の類似性について
- 切り替える回路は存在するか?
- ALUが(半分)完成!
- 次回やることはシフト回路

# 3分クッキング～補数の作り方～

## 1の補数

見た目：あるN桁の数Xに足すと、N桁の「 $111\dots111$ 」になるような数Bのこと

## 2の補数

見た目：あるN桁数Xに足すと、(N+1)桁の「 $100\dots000$ 」になるような数Bのこと

# 3分クッキング～補数の作り方～

## 1の補数

見た目：あるN桁の数Xに足すと、N桁の「*111...111*」になるような数Bのこと

作り方1：全部のビットを反転

作り方2：111...111とXOR

## 2の補数

見た目：あるN桁数Xに足すと、(N+1)桁の「*100...000*」になるような数Bのこと

作り方1：全部のビットを反転して、1を足す

作り方2：111...111とXORして、1を足す

# 補数変換回路

## 1の補数

作り方

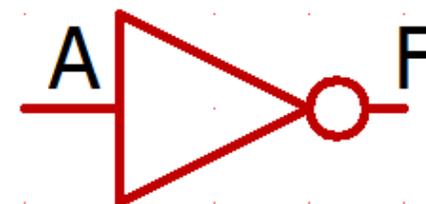
全部のビットを反転

# 補数変換回路

## 1の補数

作り方

全部のビットを反転



# 2の補数

作り方

全部のビットを反転して、1を足す

# 2の補数

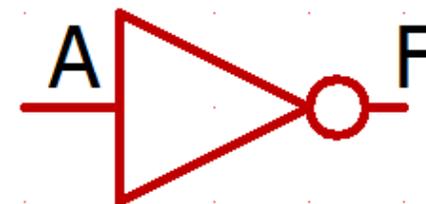
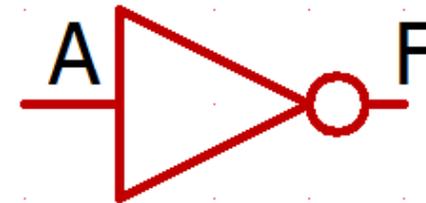
作り方

**全部のビットを反転して、1を足す**

# 2の補数

作り方

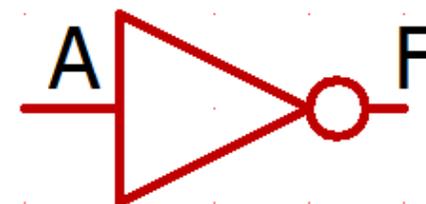
全部のビットを反転して、1を足す



# 2の補数

作り方

全部のビットを反転して、**1**を足す



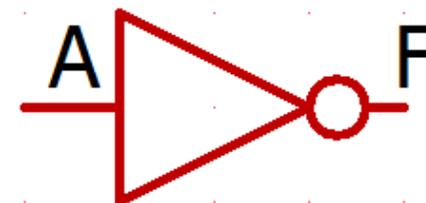
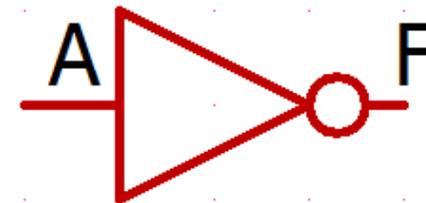
# 2の補数

作り方

全部のビットを反転して、**1を足す**

1を足すということは.....

全加算器の出番！



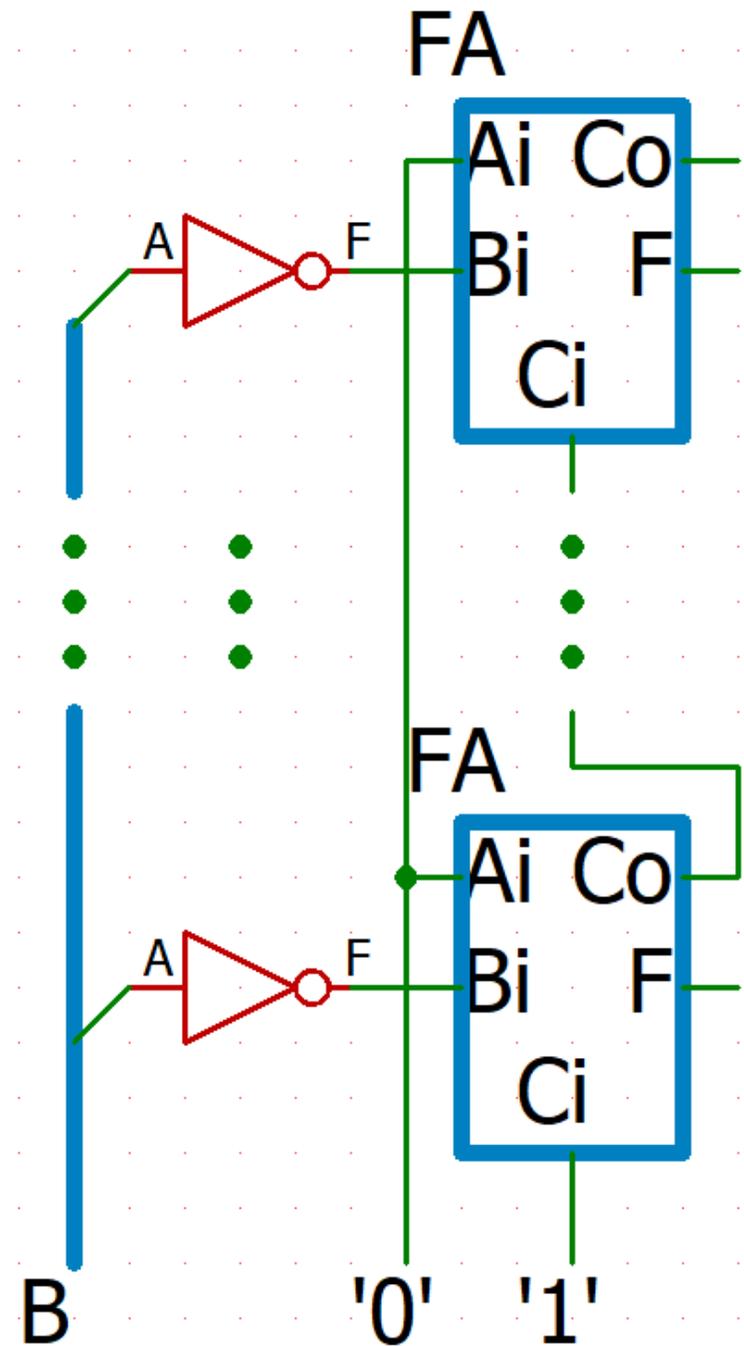
# 2の補数

作り方

全部のビットを反転して、**1を足す**

1を足すということは.....

全加算器の出番！



# 2の補数

作り方

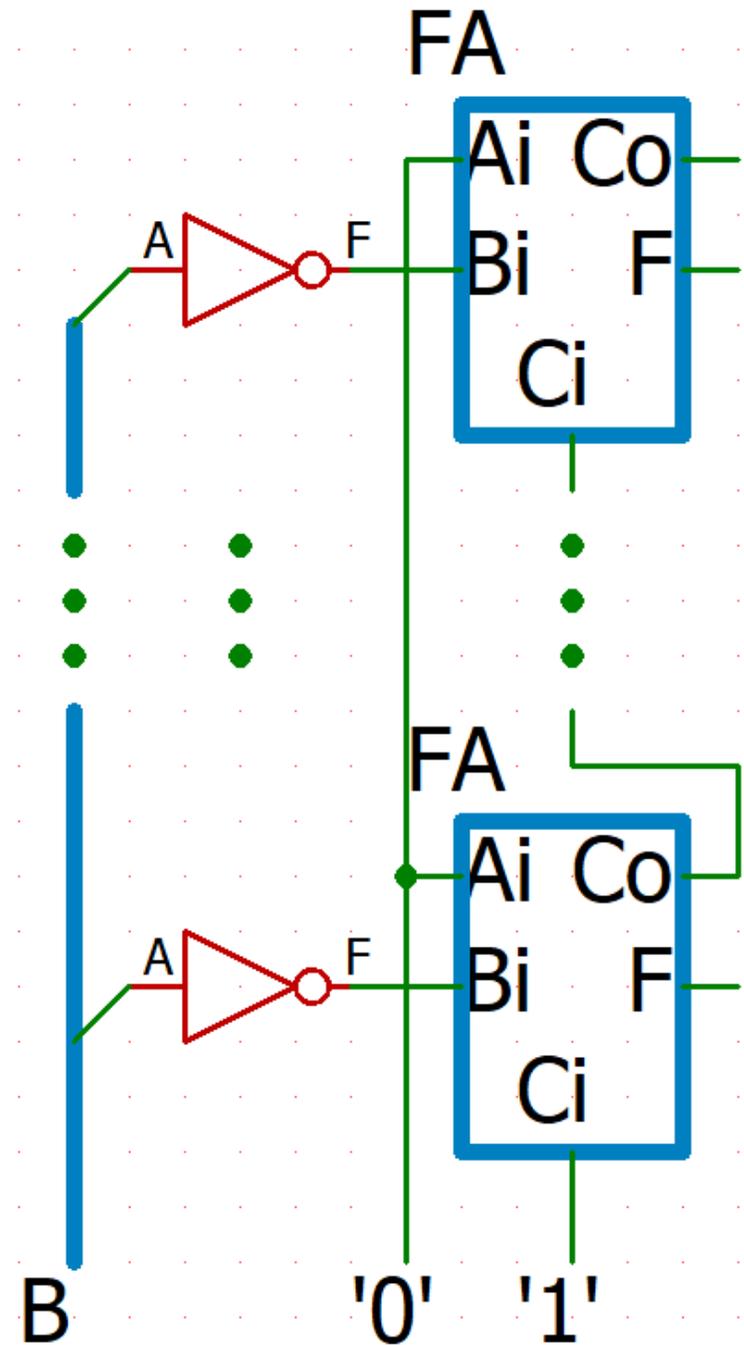
全部のビットを反転して、**1を足す**

1を足すということは.....

全加算器の出番！

全加算器があれば

補数変換もできる！



# 今日やること

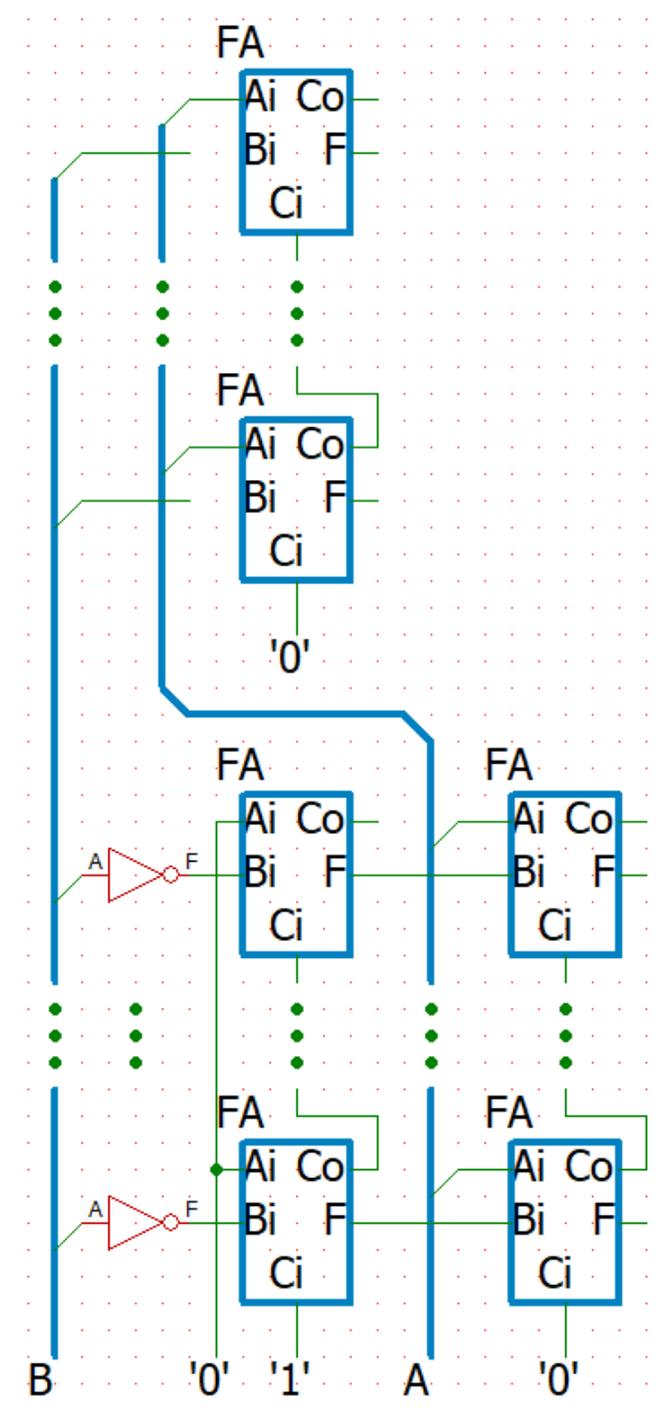
- 復習1 : 全加算器
- 復習2 : 二進数での減算
- 補数変換回路
- **補数変換回路と全加算器の類似性について**
- 切り替える回路は存在するか?
- ALUが(半分)完成!
- 次回やることはシフト回路

# 補数変換回路と

# 全加算器の類似性

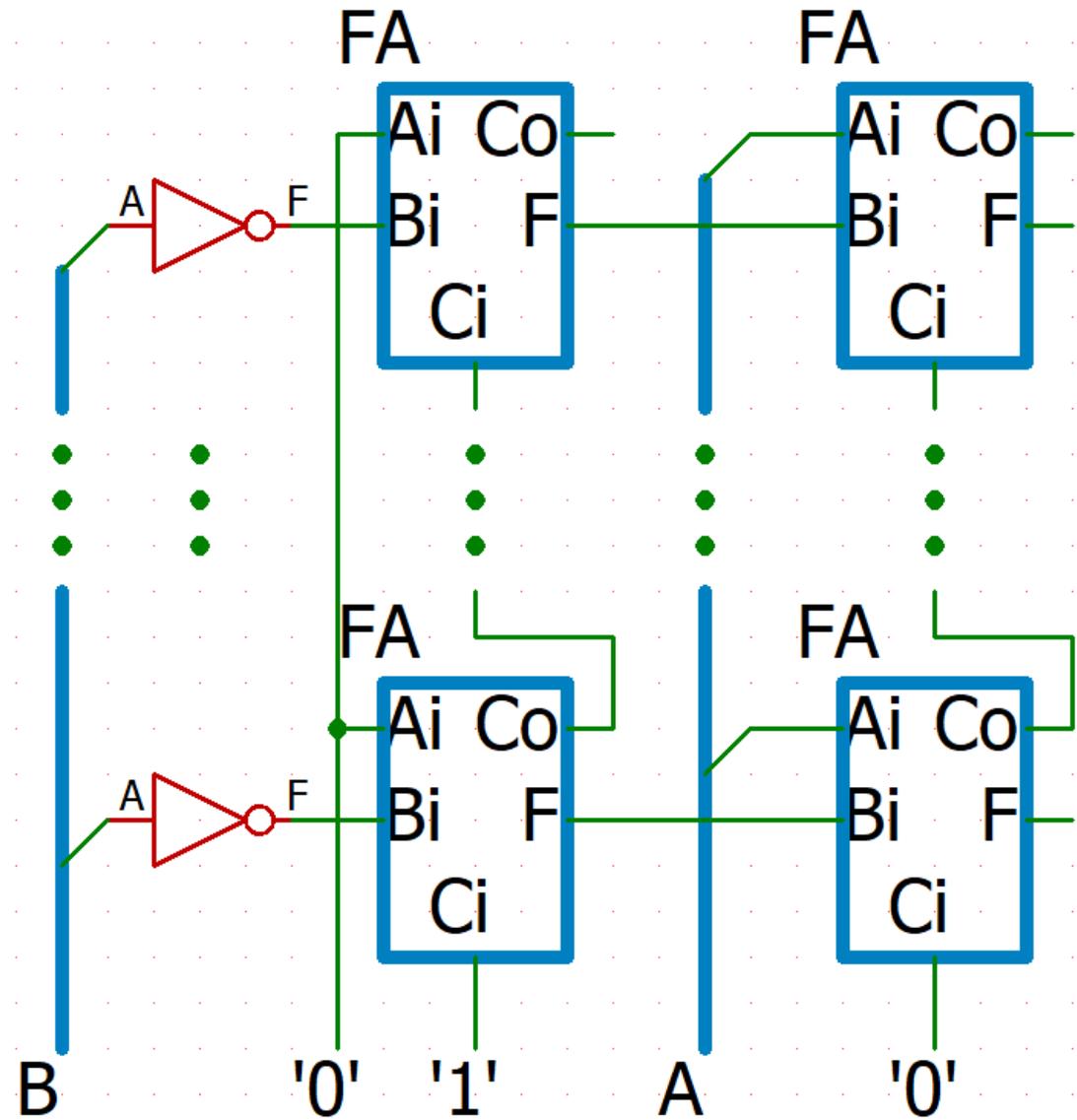
全加算器で加算ができる  
補数変換もするけど.....  
減算もやろうとすると  
全加算器が3つ必要!?

このままでは冗長な気がしませんか?  
どうにかして同じ部分を圧縮したい



# 圧縮

- Q. 補数変換回路と減算回路を圧縮したい  
 A. そうだねえ



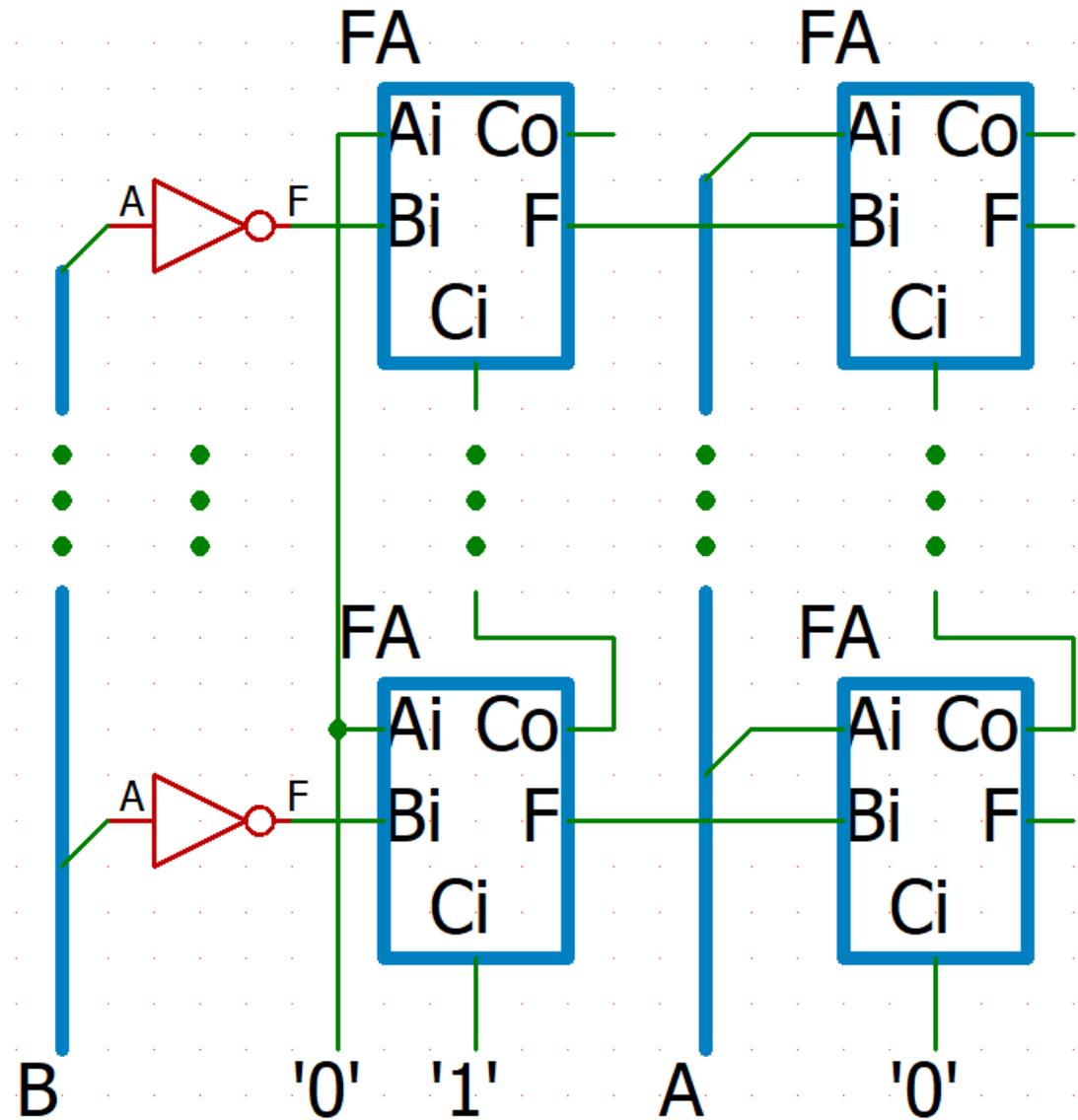
# 圧縮

Q. 補数変換回路と減算回路を圧縮したい

A. そうだねえ

Q. 補数変換回路って入力0は冗長じゃない?

A. そうだねえ



# 圧縮

Q. 補数変換回路と減算回路を圧縮したい

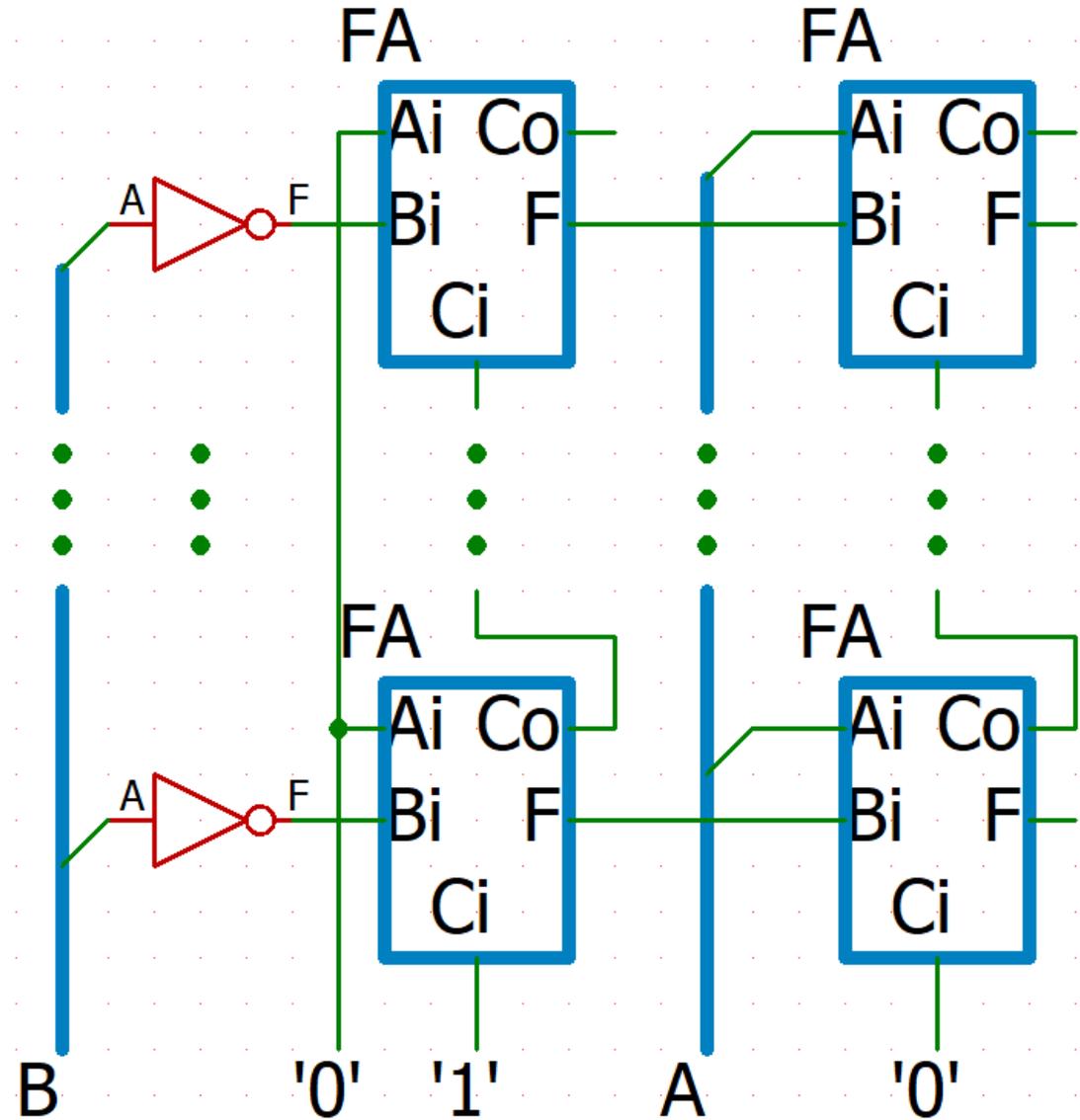
A. そうだねえ

Q. 補数変換回路って入力0は冗長じゃない?

A. そうだねえ

Q. 減算回路の1, 2段目を合体できない?

A. そうだねえ



# 圧縮

Q. 補数変換回路と減算回路を圧縮したい

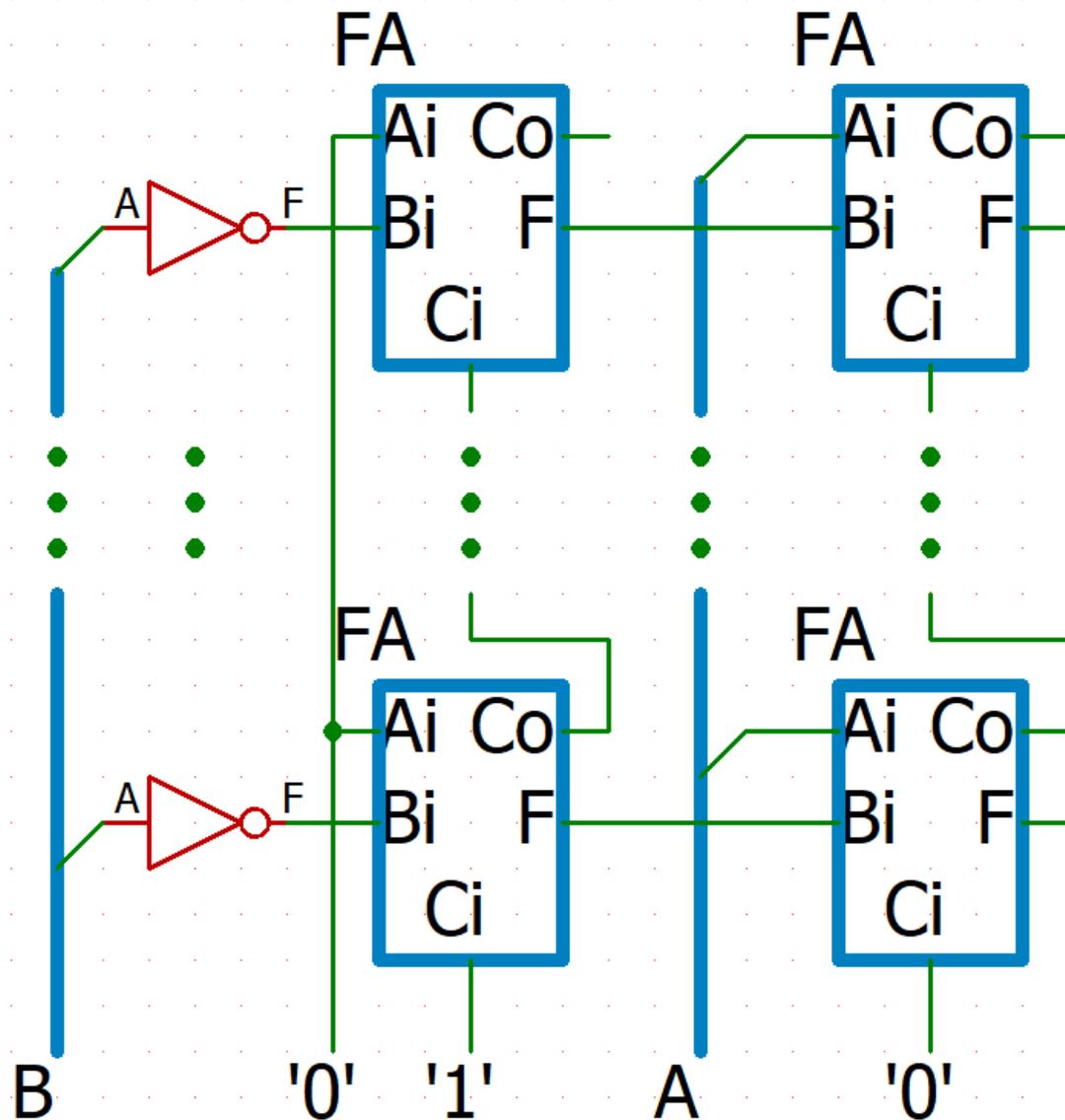
A. そうだねえ

Q. 補数変換回路って入力0は冗長じゃない?

A. そうだねえ

Q. 減算回路の1, 2段目を合体できない?

A. そうだねえ、一緒に考えてみようか



# 証明

$$(0 + \overline{B} + 1) + A = (\overline{B} + 1) + A$$

証明できたので圧縮してヨシ!

# 証明

$$(0 + \overline{B} + 1) + A = (\overline{B} + 1) + A$$

証明できたので圧縮してヨシ!

※時間が足りなかったなので簡単な証明だけしてしまいました反省しています

※編集版へ、ここカットをお願いします

# 圧縮

Q. 補数変換回路と減算回路を圧縮したい

A. そうだねえ

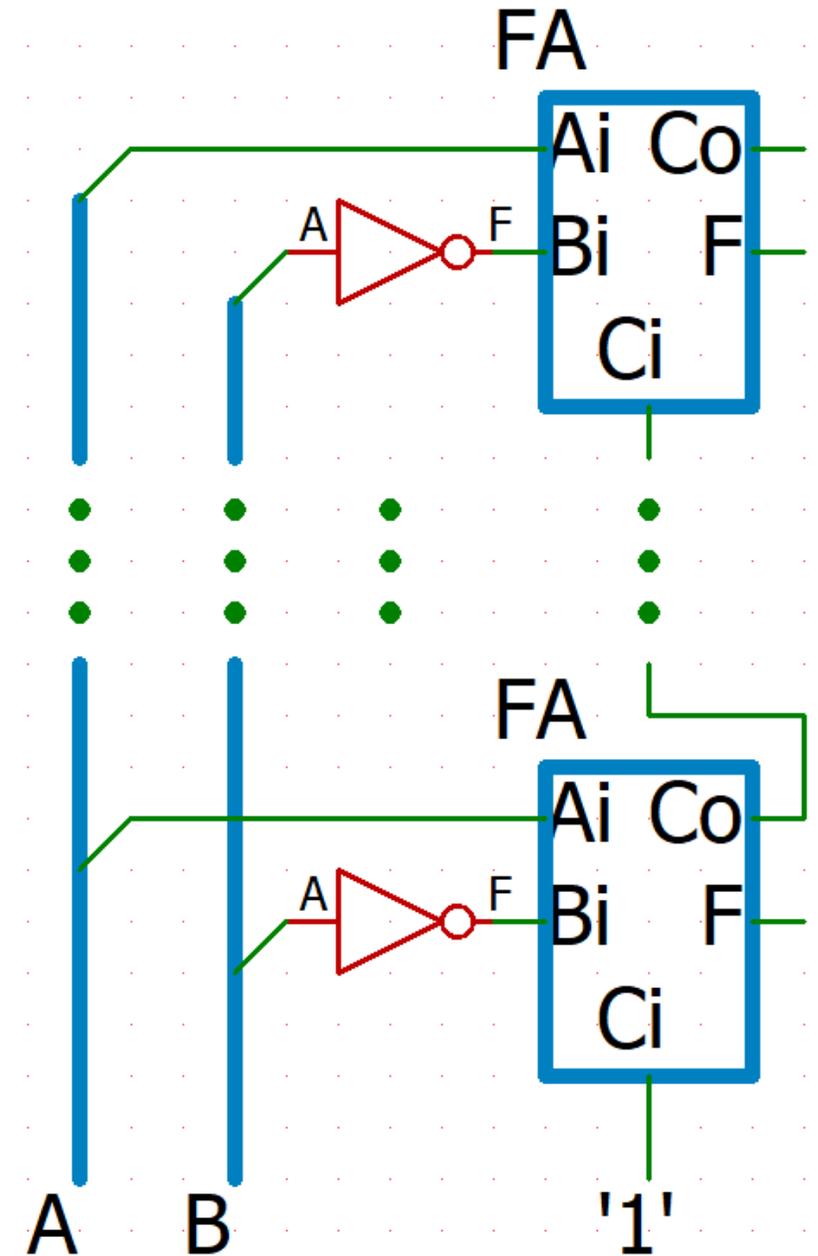
Q. 補数変換回路って入力0は冗長じゃない?

A. そうだねえ

Q. 減算回路の1, 2段目を合体できない?

A. そうだねえ、一緒に考えてみようか

証明できたのでこれをこうしてこうじゃ  
ちなみにこれは**全減算器**といいます



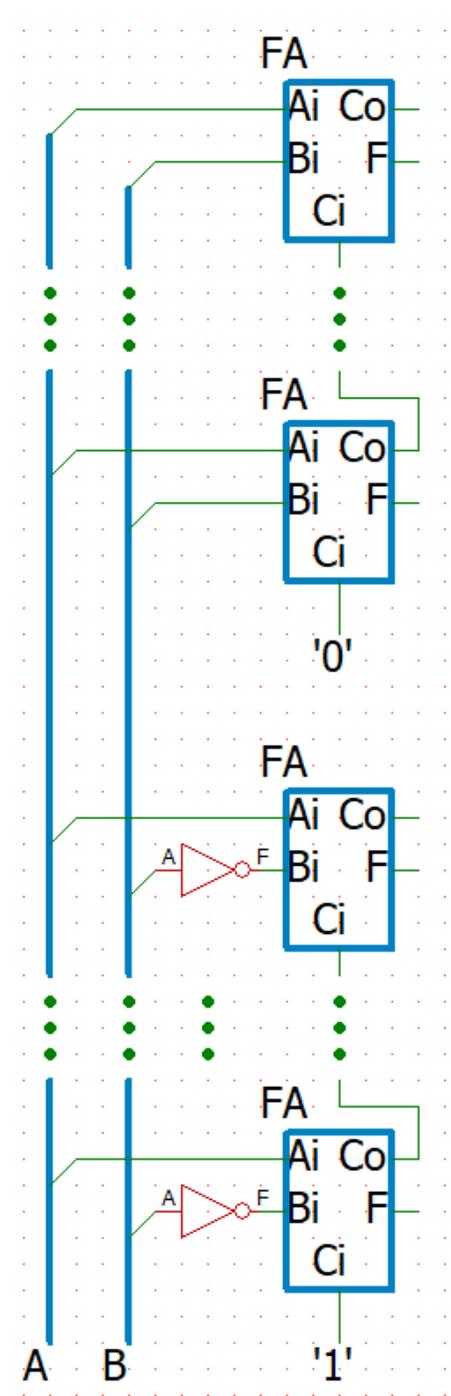
# 今日やること

- 復習1：全加算器
- 復習2：二進数での減算
- 補数変換回路
- 補数変換回路と全加算器の類似性について
- **切り替える回路は存在するか?**
- ALUが(半分)完成!
- 次回やることはシフト回路

# 切り替える回路は存在するか?

Q. まだ圧縮したい

A. そうだねえ



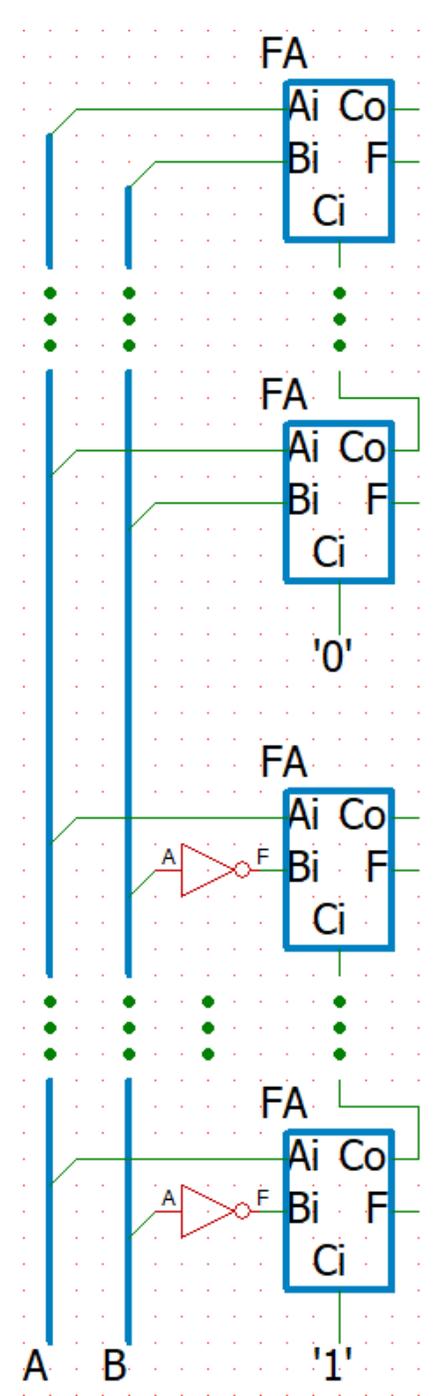
# 切り替える回路は存在するか？

Q. まだ圧縮したい

A. そうだねえ

Q. 全加算器と全減算器って、  
入力を切り替えてるだけでほぼ一緒じゃない？

A. そうだねえ



# 切り替える回路は存在するか?

Q. まだ圧縮したい

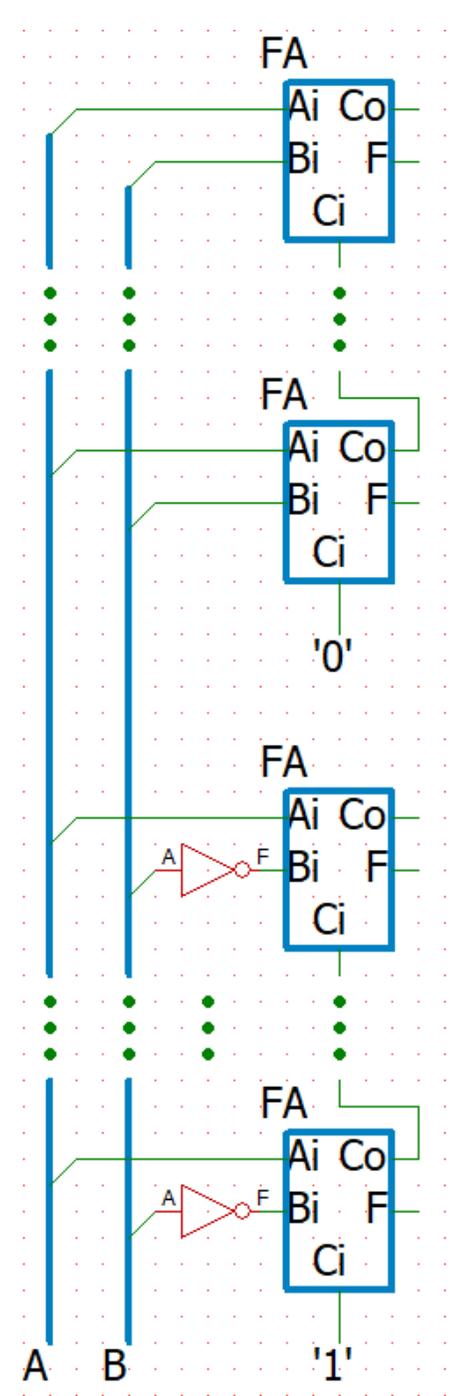
A. そうだねえ

Q. 全加算器と全減算器って、  
入力を切り替えてるだけでほぼ一緒じゃない?

A. そうだねえ

Q. 加算減算を区別するフラグがあるとして、  
それに応じて切り替える回路は存在しないの?

A. そうだねえ



# 切り替える回路は存在するか？

Q. まだ圧縮したい

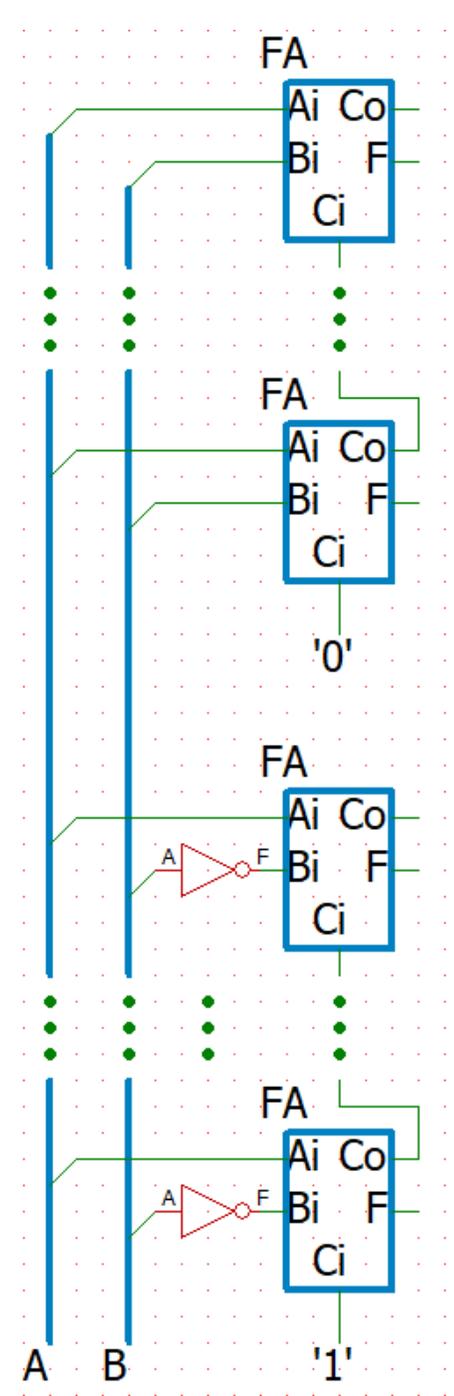
A. そうだねえ

Q. 全加算器と全減算器って、  
入力を切り替えてるだけでほぼ一緒じゃない？

A. そうだねえ

Q. 加算減算を区別するフラグがあるとして、  
それに応じて切り替える回路は存在しないの？

A. そうだねえ、一緒に考えてみようか



# 切り替え回路は存在するか？

## 要求

入力A, B, Cと出力Sがあり

入力Cが0のとき出力はA、入力Cが1のとき出力がBとなる論理回路

## 考え方

- ① 入力Cが0のとき出力がA、入力Cが1のとき出力が0
- ② 入力Cが0のとき出力が0、入力Cが1のとき出力がB
- ③ ①と②の和

# ① 入力Cが0のとき出力がA、入力Cが1のとき出力が0

A	C	S_1
0	0	0
0	1	0
1	0	1
1	1	0

A	C	S_1
?	0	A
?	1	0

? := 0でも1よい  
(don't care)

左の表をまとめると

$$S_1 = A\bar{C}$$

これで

- 入力Cが0のとき出力がA
- 入力Cが1のとき出力が0

の2つの特性を満たす回路が得られた

※補足

左の表は更に右の表に圧縮できる

## ② 入力Cが0のとき出力が0、入力Cが1のとき出力がB

A	C	S_2
0	0	0
0	1	0
1	0	0
1	1	1

A	C	S_2
?	0	0
?	1	B

? := 0でも1よい  
(don't care)

左の表をまとめると  
 $S_1 = BC$

これで

- 入力Cが0のとき出力が0
- 入力Cが1のとき出力がB

の2つの特性を満たす回路が得られた

※補足

左の表は更に右の表に圧縮できる

### ③ : ①と②の和

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

A	B	C	S
?	?	0	A
?	?	1	B

? := 0でも1よい  
(don't care)

左の表をまとめると  
 $S = A\bar{C} + BC$

これで

- 入力Cが0のとき出力がA
- 入力Cが1のとき出力がB

の2つの特性を満たす回路が得られた

※補足

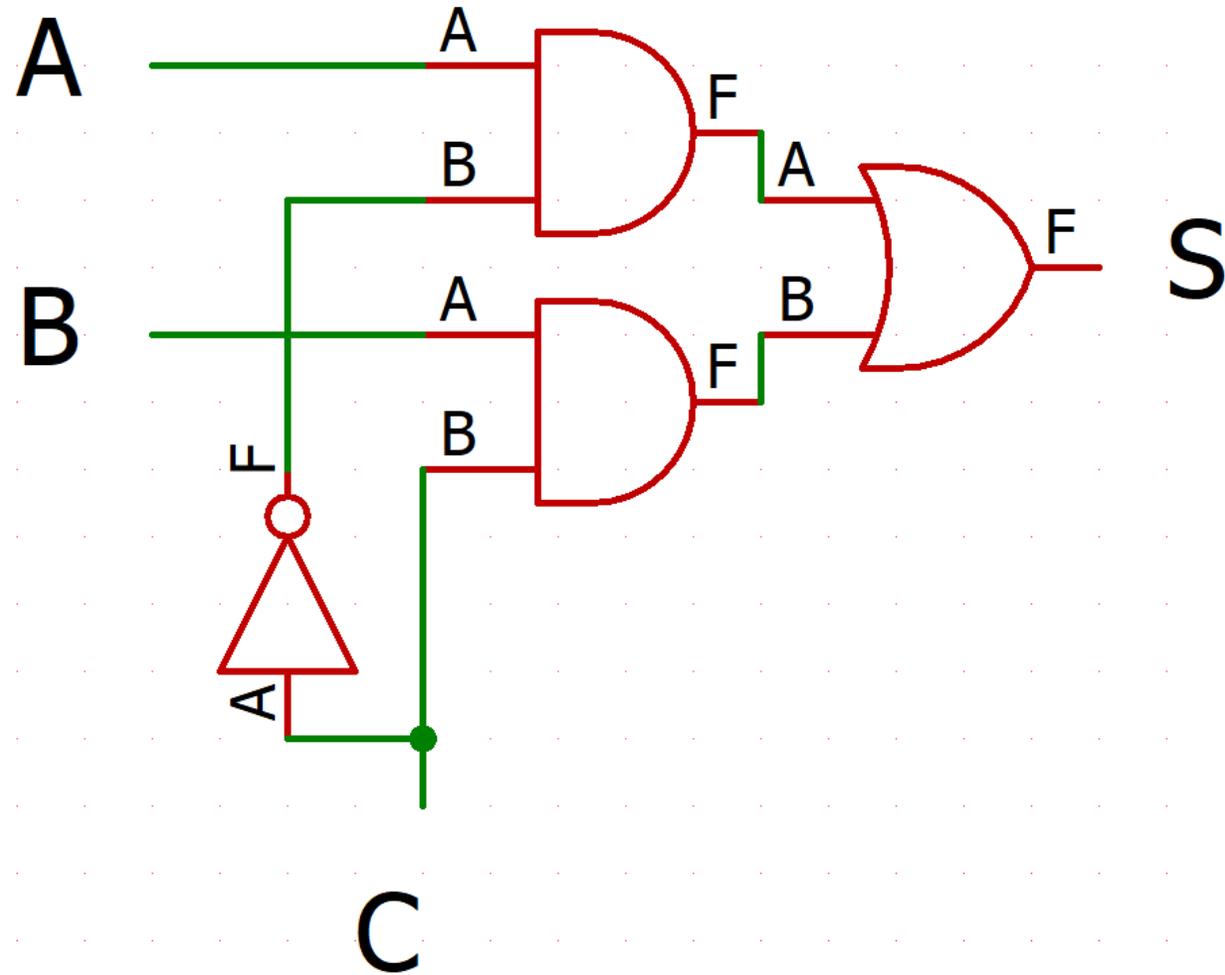
左の表は更に右の表に圧縮できる

# 切り替え回路は存在するか？

Q. 圧縮できたね

A. そうだねえ、一緒に示したね

ちなみにこれは**セレクタ**といいます  
 多分、次々々々回ぐらいに出てくるので  
 頭の片隅に置いて下さい



# 今日やること

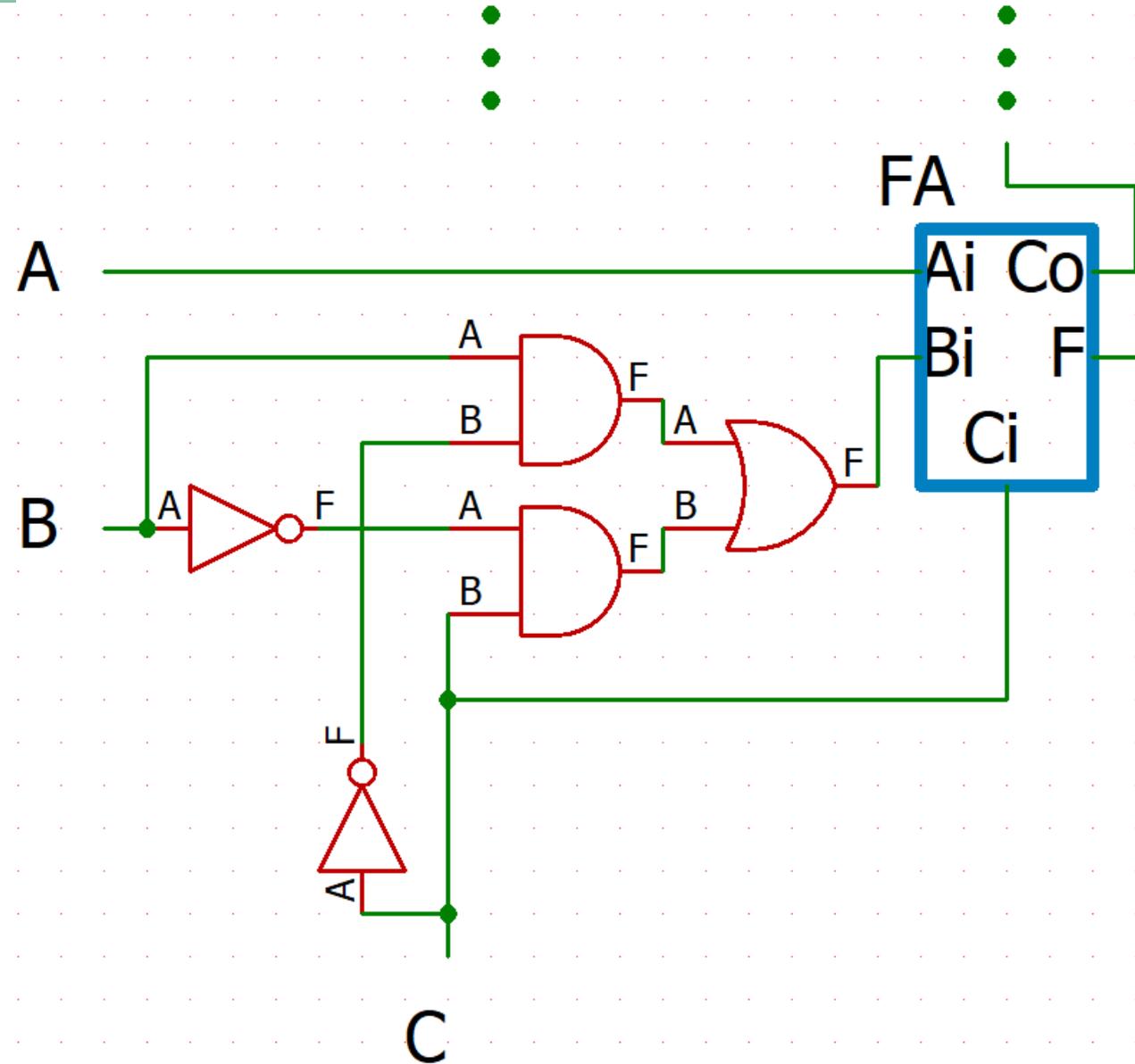
- 復習1 : 全加算器
- 復習2 : 二進数での減算
- 補数変換回路
- 補数変換回路と全加算器の類似性について
- 切り替える回路は存在するか?
- **ALUが(半分)完成!**
- 次回やることはシフト回路

# ALUが(半分)完成!

- 全加算器と全減算器をセレクトで合体!
- セレクトを圧縮できないか?
- ALUが(半分)完成!

# セレクトで合体!

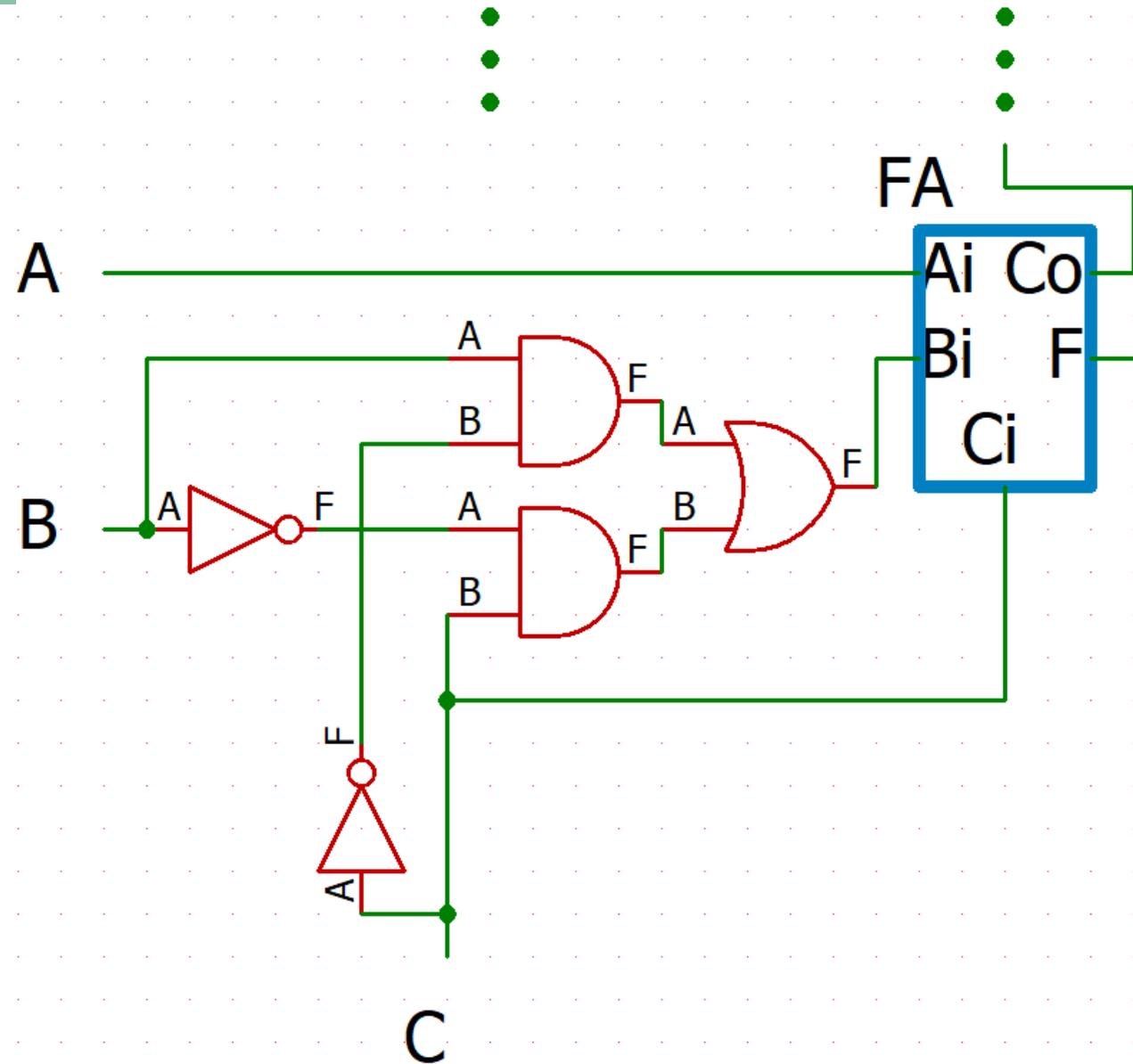
全加算器の入力段にセレクトを設け、  
加算と減算を切り替えれるようになった



# 圧縮できない？

Q. このセレクトってゴツくない？

A. そうだねえ





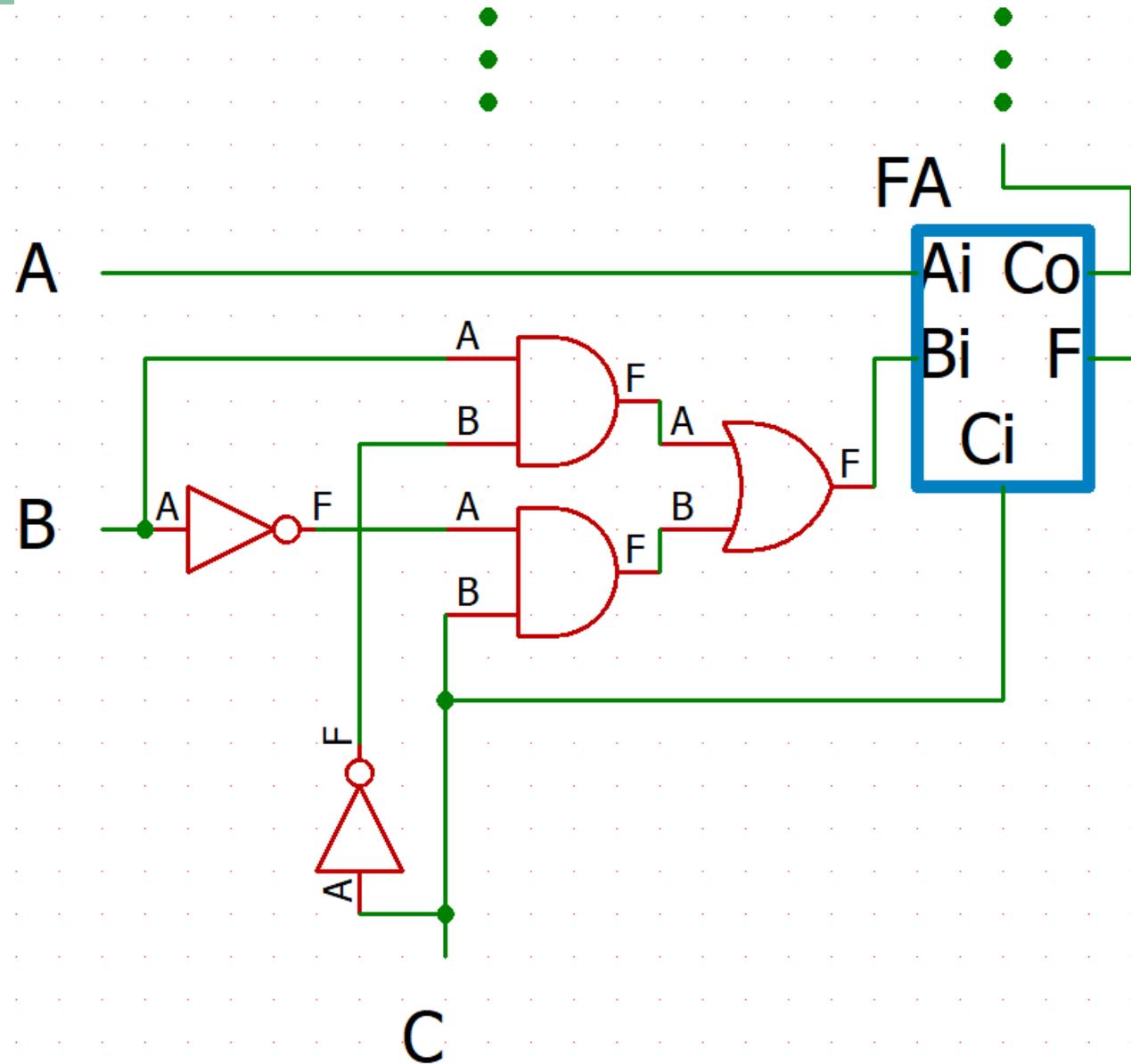
# 圧縮できない？

Q. このセレクトってゴツくない？

A. そうだねえ

Q. このセレクトって圧縮できない？

A. そうだねえ、一緒に考えてみようか



# セレクタ圧縮

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Q. Aが0のときは出力が $B$ 、  
Aが1のときは出力が $\bar{B}$   
の回路ってなーんだ

A. XOR

# 圧縮できない?

Q. このセレクトってゴツくない?

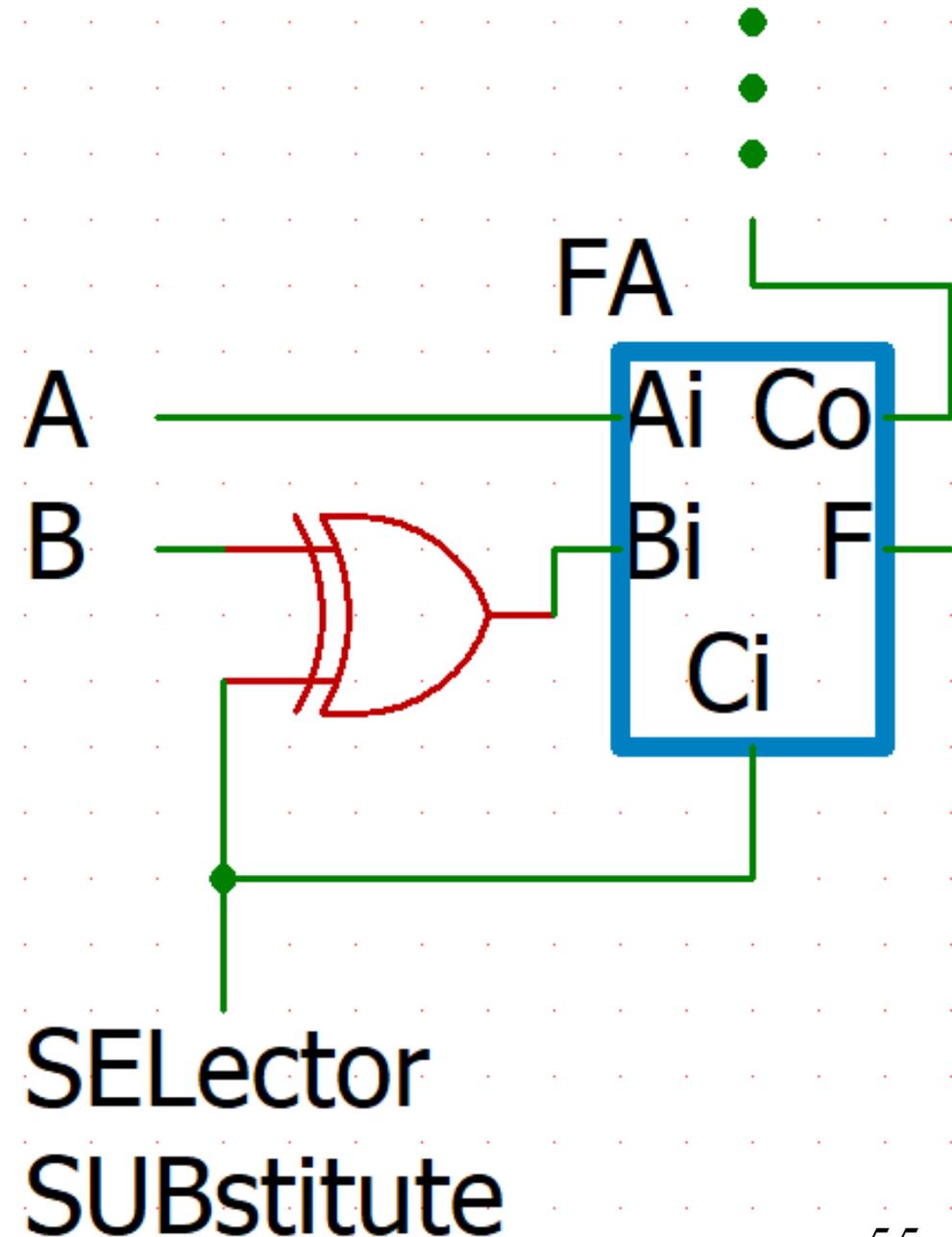
A. そうだねえ

Q. このセレクトって圧縮できない?

A. そうだねえ、一緒に考えてみようか

Q. 圧縮できたね

A. そうだねえ、一緒に示したね

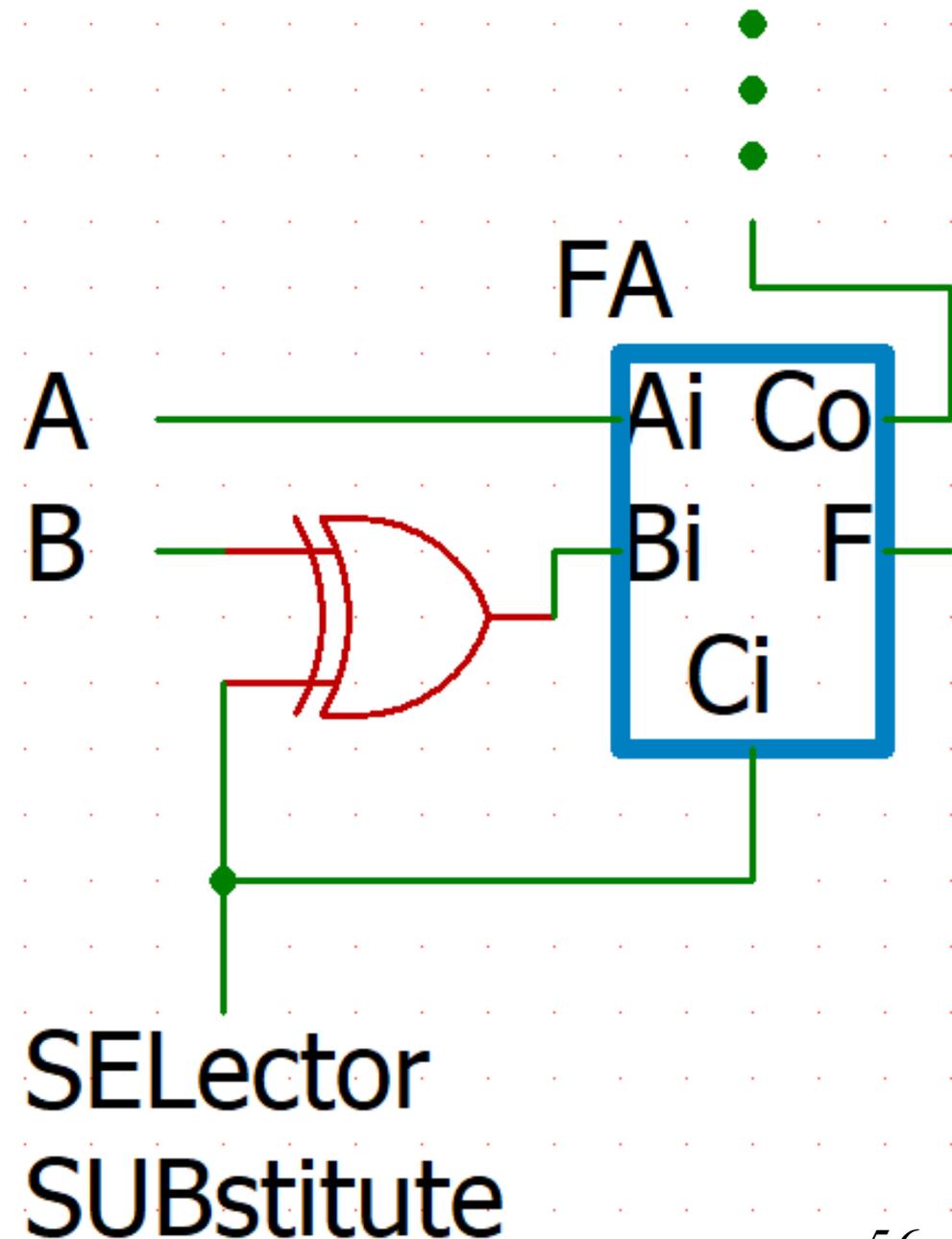


# ALUが(半分)完成!

ようやく、算術回路の主要部分が完成!  
 (符号による大小判定が残ってるけど後回し)

CS集会 #8 #9 #10 #11 を終えてようやく!

長かった.....



# 今日やること

- 復習1 : 全加算器
- 復習2 : 二進数での減算
- 補数変換回路
- 補数変換回路と全加算器の類似性について
- 切り替える回路は存在するか?
- ALUが(半分)完成!
- **次回やることはシフト回路**

# 次回

- シフト回路
- 右シフト
- 左シフト

新しく出てくる計算手法はこれだけ  
シフト回路をやれば実質Logic部分終わり  
次回でようやくALUが完成する見込み

# 今日やったこと1

- 復習1：全加算器
  - 半加算器2つで全加算器
  - 並列に並べてNビット全加算器
- 復習2：二進数での減算
  - 1の補数は反転するだけ
  - 2の補数は反転して1を足す
- 補数変換回路
  - NOTで反転
  - 1を足す
  - 全減算器できた

# 今日やったこと2

- 全減算器と全加算器の類似性について
  - FAが並列に並んでいるのは同じ
  - 並列に並べたくない
  - 入力部分を切り替えたい
  - この切替回路は存在するか?
- 切り替える回路は存在するか?
  - 実はこれ1bitセレクタ
- ALUが(半分)完成!
  - 加算回路
  - 加算 / 減算切替回路

# 次回

- シフト回路
- 右シフト / 左シフト