

!急募!

今日の動画撮影

してくれる人

CS 集会

2

今日からお前も計算機

自己紹介



Twitter@yonabeyona

- 名前 : 夜鍋 ヨナ (よなべ よな)
- Twitter : yonabeyona
- Discord : yona_47
- その他 : CS好き、数学勉強中
: CSの中でもCAあたりが専門

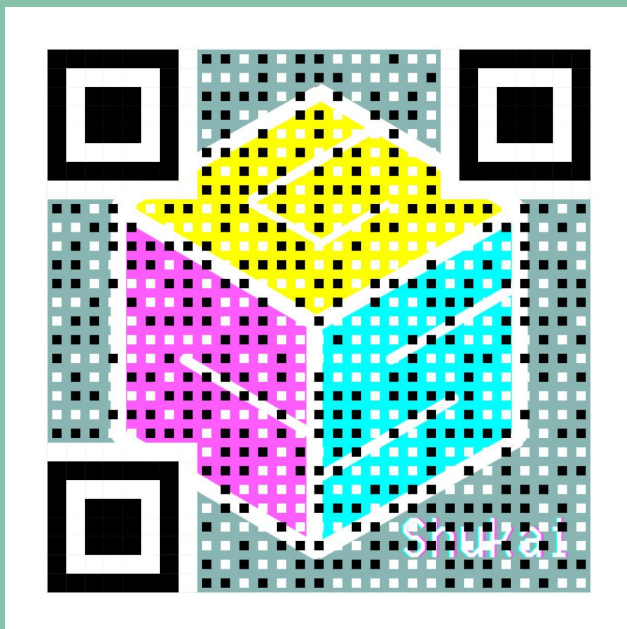
今日やること

- 告知
- これも立派な計算機
- NOTがあれば何でもできる
- 今日からお前も計算機
- 前回の質問回答

今日やること

- 告知
- これも立派な計算機
- NOTがあれば何でもできる
- 今日からお前も計算機
- 前回の質問回答

CS 集会 告知



discord.gg/ZCnUCmncDK

Discord

サーバー作りました

1. #rules 読んで
2. #自己紹介したら
3. ロールをもらって
4. 全部見よう
5. Discord運営募集中

今日やること

- 告知
- これも立派な計算機
- NOTがあれば何でもできる
- 今日からお前も計算機
- 前回の質問回答

木材トランジスタ

- 軽くて丈夫なバルサ材という木材を利用
- リグニンを除去、PEDOT:PSSを添加している
- 導電率 69[S/m] ← 純粋なゲルマニウム(半導体)ぐらい
- 今までの研究と異なり、継続的に使うことができる
- オンオフのスイッチングができる → コンピュータが構成できる
- <https://gigazine.net/news/20230502-wooden-transistor/>

光論理ゲート

- 3入力、1出力で論理演算を行う「 Ψ ゲート」をNTTが開発
- CMOSによる論理ゲートよりも遅延が $1/300$ で早い
- 1つの素子で複数の演算を行うことができる
- 論理演算ができる \rightarrow コンピュータが構成できる
- 様々な波長で同時に異なる演算ができる \leftarrow ここWDM
- <https://xtech.nikkei.com/atcl/nxt/mag/ne/18/00001/00129/>

リレー式ゲート

- 電磁石による電気接点のオンオフを切り替える
- オンオフのスイッチングができる → コンピュータが構成できる
- 大昔のコンピュータはリレー式コンピュータだった
- 日本の富士通も作成してた。池田敏雄(1923-1974)さんが有名。
- 池田敏雄さんの話はプロジェクトX第1期で紹介されてる

今日やること

- 告知
- これも立派な計算機
- **NOTがあれば何でもできる**
- 今日からお前も計算機
- 前回の質問回答

NANDだけ

使って作ろう

論理回路

夜鍋ヨナ





NOT

~~NAND~~だけ

使って作ろう

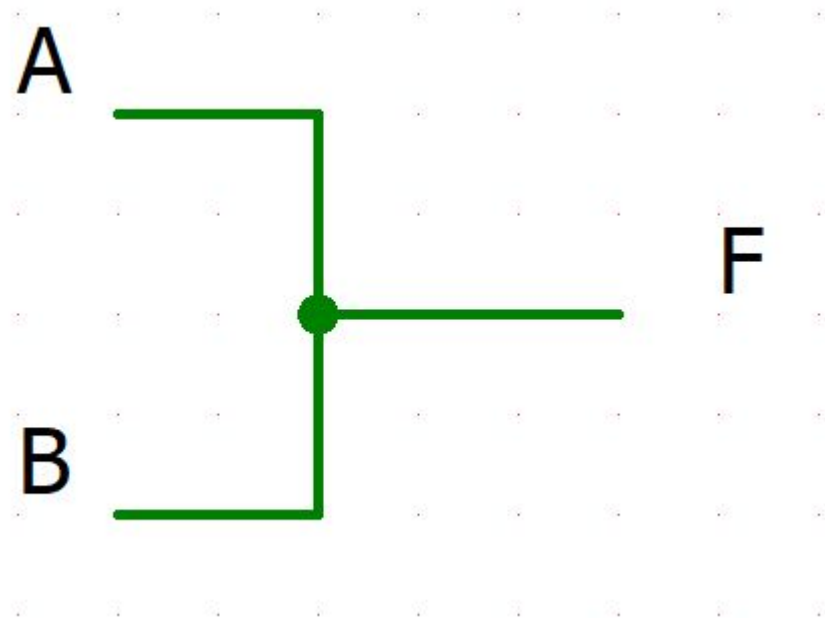
論理回路

夜鍋ヨナ

NOTがあれば何でもできるの前に

特殊なORの話

$A=0、B=0 \rightarrow F=0$

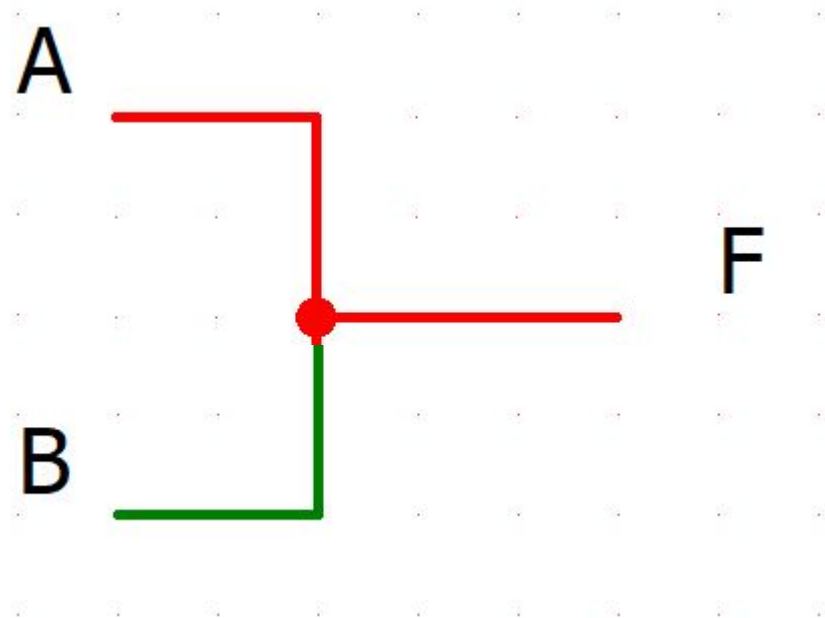


NOTがあれば何でもできるの前に

特殊なORの話

$A=0, B=0 \rightarrow F=0$

$A=1, B=0 \rightarrow F=1$



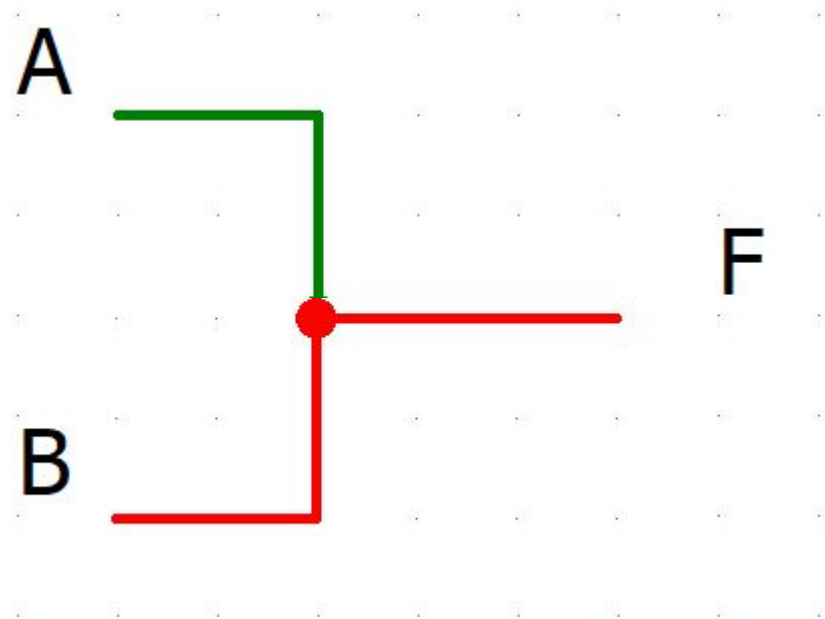
NOTがあれば何でもできるの前に

特殊なORの話

$A=0, B=0 \rightarrow F=0$

$A=1, B=0 \rightarrow F=1$

$A=0, B=A \rightarrow F=1$



NOTがあれば何でもできるの前に

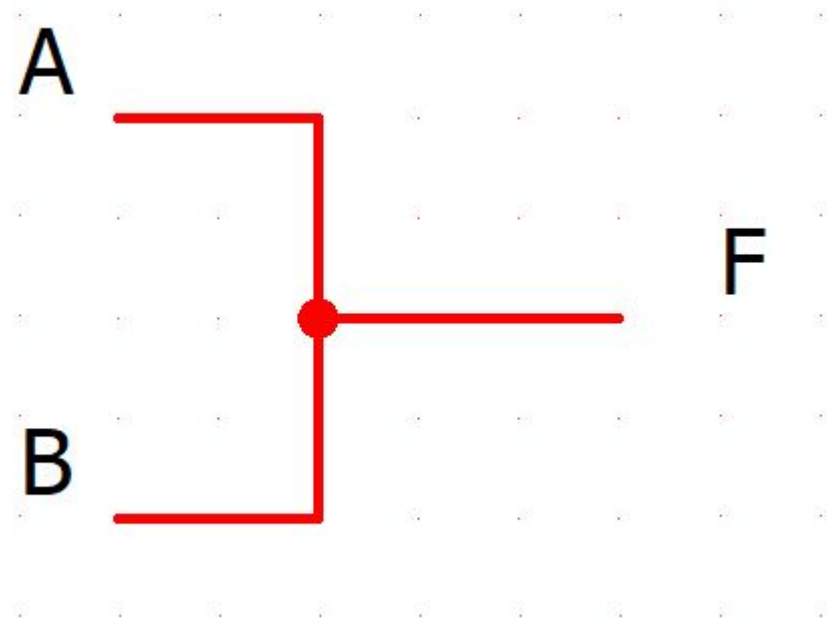
特殊なORの話

$A=0, B=0 \rightarrow F=0$

$A=1, B=0 \rightarrow F=1$

$A=0, B=A \rightarrow F=1$

$A=1, B=1 \rightarrow F=1$

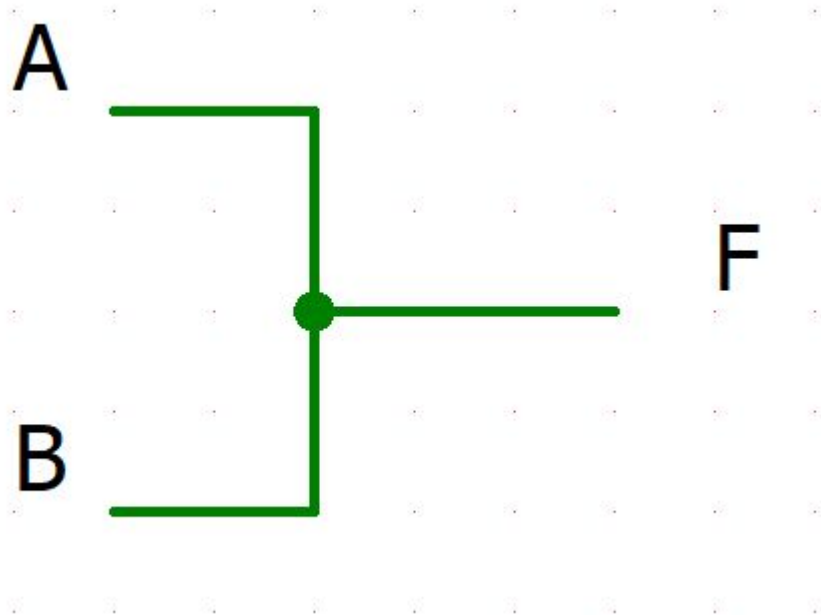


NOTがあれば何でもできるの前に

特殊なORの話

名前はワイヤードオア(Wired OR)

A	B	F
0	0	0
1	0	1
0	1	1
1	1	1

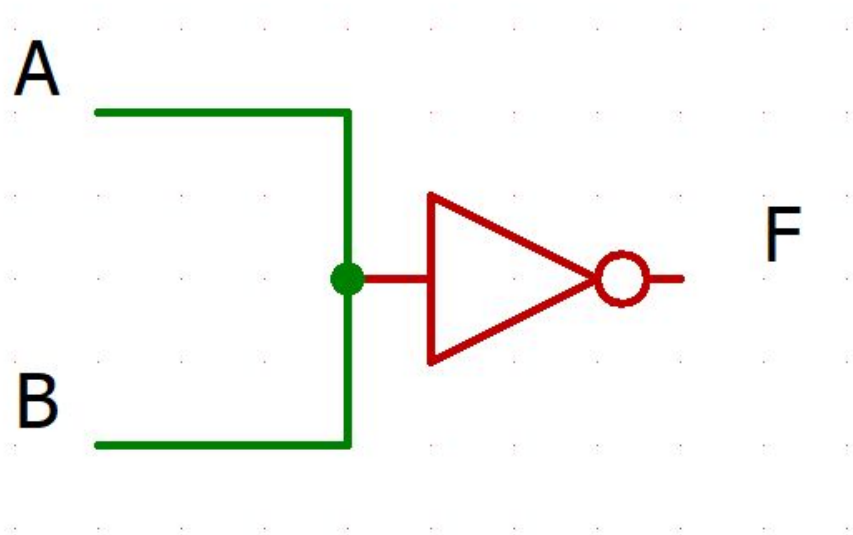


NOR は NOT の入力をワイヤードオア

真理値表

A	B	F
0	0	1
1	0	0
0	1	0
1	1	0

真理値表

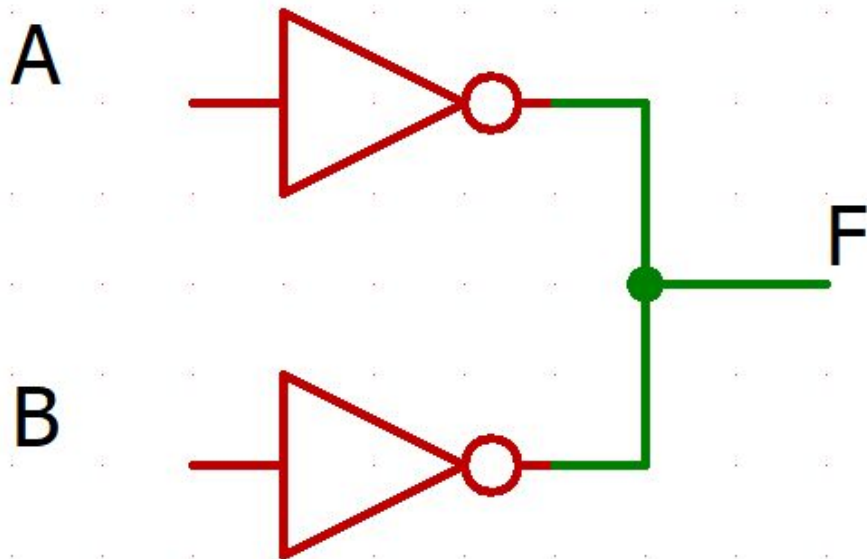


NAND は NOT の出力をワイヤードオア

真理値表

A	B	F
0	0	1
1	0	1
0	1	1
1	1	0

真理値表



NOTがあれば何でもできる Q&A

Q1 : AからB、BからAにも電流流れるよね?逆流しない?

A1 : 現実的にはダイオードを入れて一方通行にする

Q2 : 論理ゲートを電気素子で作る方法の話は?

A2 : DTL(Diode-Transistor Logic)、TTL(Transistor-Transistor Logic)

CMOS(Complementary Metal-Oxide-Semiconductor)で作る話がある

後日のネタでやるつもり、CPU設計の本筋から外れるので後々になる

今日やること

- 告知
- これも立派な計算機
- NOTがあれば何でもできる
- 今日からお前も計算機
- 前回の質問回答

今日からお前も計算機

- NOTがあれば何でもできる
- 光でも木でもリレーでもNOTを構築できる → コンピューターになる

今日からお前も計算機

- NOTがあれば何でもできる
- 光でも木でもリレーでもNOTを構築できる → コンピューターになる
- 否定するものは論理演算ができる
- 否定さえできれば、何でもコンピューターになる

今日からお前も計算機

- NOTがあれば何でもできる
- 光でも木でもリレーでもNOTを構築できる → コンピューターになる
- 否定するものは論理演算ができる
- 否定さえできれば、何でもコンピューターになる
- オタクはすぐ否定する(否定から入る)

今日からお前も計算機

- NOTがあれば何でもできる
- 光でも木でもリレーでもNOTを構築できる → コンピューターになる
- 否定するものは論理演算ができる
- 否定さえできれば、何でもコンピューターになる
- オタクはすぐ否定する(否定から入る)
- オタクで論理演算ができるので、お前も論理演算ができる

例「最強のエディタは、VimでもEmacsでもない」

- 用意するもの：オタク1人
- 手法：オタクを用意して、 $A+B$ を質問する
- 入力： $A :=$ 最強のエディタはVim、 $B :=$ 最強のエディタはEmacs
- 出力：F

例

- 最強のエディタは、VimでもEmacsでももない

A: 最強のエディタは Vim?

B: 最強のエディタは Emacs?

いや、違うが?



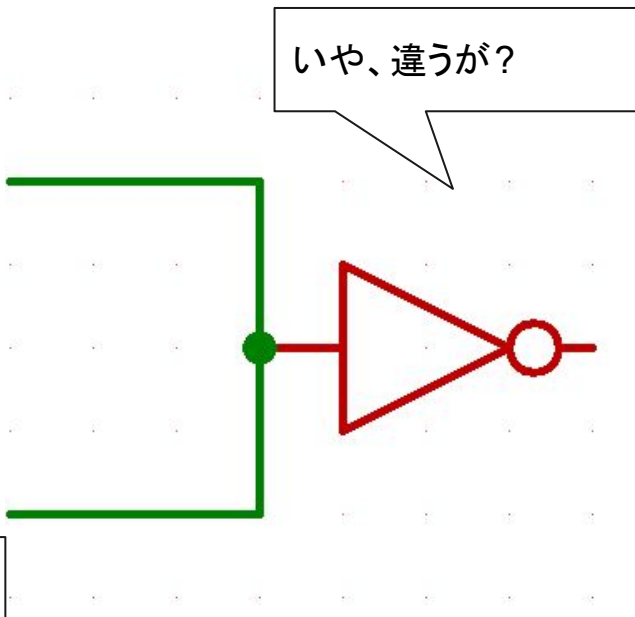
例

- 最強のエディタは、VimでもEmacsでももない

A: 最強のエディタは Vim?

B: 最強のエディタは Emacs?

いや、違うが?



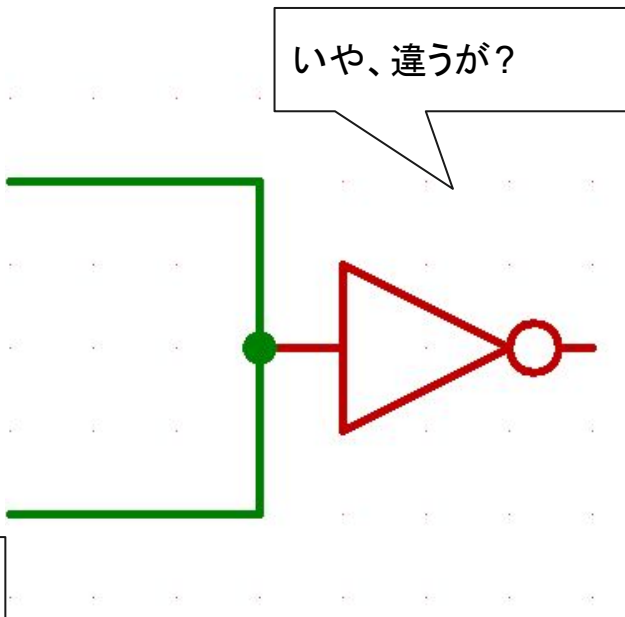
例

- 最強のエディタは、VimでもEmacsでももない

A: 最強のエディタは Vim?

いや、違うが?

B: 最強のエディタは Emacs?



$$F = \overline{A + B}$$

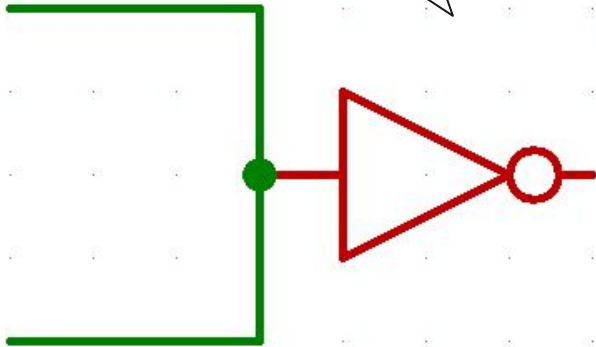
例

- 最強のエディタは、VimでもEmacsでももない

A: 最強のエディタは Vim?

B: 最強のエディタは Emacs?

いや、違うが?



F: 最強のエディタは VimでもEmacsでもない

今日からお前も計算機

- NOTがあれば何でもできる
- 光でも木でもリレーでもNOTを構築できる → コンピューターになる
- 否定するものは論理演算ができる
- 否定さえできれば、何でもコンピューターになる
- オタクはすぐ否定する(否定から入る)
- オタクで論理演算ができるので、お前も論理演算ができる

今日やること

- 告知
- これも立派な計算機
- NOTがあれば何でもできる
- 今日からお前も計算機
- 前回の質問回答

前回の質問回答

Q1. XOR, XNORだけで論理回路を構成できるか?

A1. 可能。前回の内容を拡張して、TrueとFalseを導入することで解決できる。

Q2. NANDだけとNORだけロジックゲートの違いは?

A2. NANDの方が半導体設計時の実装面積に置いて有利。CMOSによるゲートの実装

と半導体の物理的特性に触れることで解説できる

前回の質問回答

Q1. XOR, XNORだけで論理回路を構成できるか？

A1. 可能。前回の内容を拡張して、TrueとFalseを導入することで解決できる。

Q2. NANDだけとNORだけロジックゲートの違いは？

A2. NANDの方が半導体設計時の実装面積に置いて有利。

CMOSによるゲートの実装と半導体の物理的特性に触れることで解説できる

Q1. XOR, XNORだけで論理回路を構成できるか？

- ①: XORでNOTを構成できる
- ②: NOTですべての論理回路を構成できる
- ③: ①と②より、XORですべての論理回路を構成できる

Q1. XOR, XNORだけで論理回路を構成できるか？

- ①: XORでNOTを構成できる
- ②: NOTですべての論理回路を構成できる
- ③: ①と②より、XORですべての論理回路を構成できる

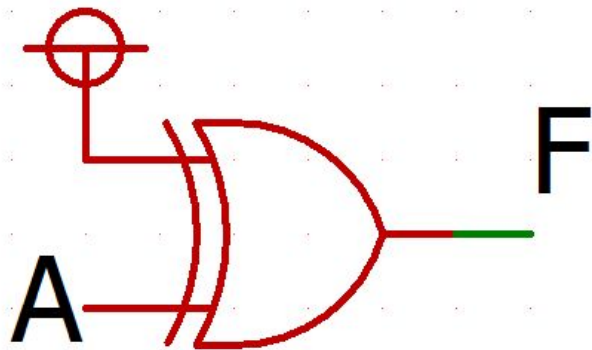
→ XOR(で作ったNOT)があれば何でもできる

Q1. XOR, XNORだけで論理回路を構成できるか？

XORでNOTを構成する

$$\begin{aligned}
 \bar{A} &= \bar{A} + 0 \\
 &= \bar{A} \bullet 1 + 0 \\
 &= \bar{A} \bullet 1 + A \bullet 0 \\
 &= \bar{A} \bullet 1 + A \bullet \bar{1} \\
 &= A \oplus 1
 \end{aligned}$$

True, High, 1



Q1. XOR, XNORだけで論理回路を構成できるか？

XORでNOTを構成する

(1) XORの真理値表

A	B	F
0	0	0
1	0	1
0	1	1
1	1	0

(2) 都合の良い所だけ選ぶ

A	B	F
0	0	0
1	0	1
0	1	1
1	1	0

(3) NOTになった

A	B	F
0	1	1
1	1	0

Q1. XOR, XNORだけで論理回路を構成できるか？

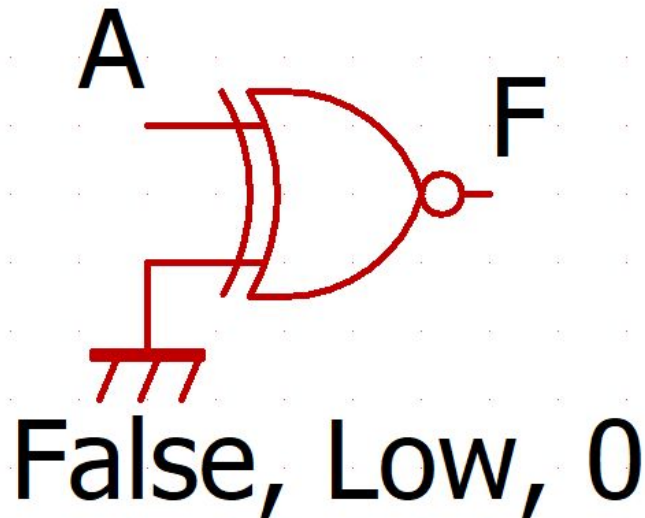
- ①: XNORでNOTを構成できる
- ②: NOTですべての論理回路を構成できる
- ③: ①と②より、XNORですべての論理回路を構成できる

→ XNOR(で作ったNOT)があれば何でもできる

Q1. XOR, XNORだけで論理回路を構成できるか？

XNORでNOTを構成する

$$\begin{aligned}
 \bar{A} &= \bar{A} + 0 \\
 &= \bar{A} \bullet 1 + 0 \\
 &= \bar{A} \bullet 1 + A \bullet 0 \\
 &= \bar{A} \bullet \bar{0} + A \bullet 0 \\
 &= \overline{A \oplus 0}
 \end{aligned}$$



Q1. XOR, XNORだけで論理回路を構成できるか？

XNORでNOTを構成する

(1)XNORの真理値表

A	B	F
0	0	1
1	0	0
0	1	0
1	1	1

(2)都合の良い所だけ選ぶ

A	B	F
0	0	1
1	0	0
0	1	0
1	1	1

(3)NOTになった

A	B	F
0	0	1
1	0	0

前回の質問回答

Q1. XOR, XNORだけで論理回路を構成できるか？

A1. 可能。前回の内容を拡張して、TrueとFalseを導入することで解決できる。

Q2. NANDだけとNORだけロジックゲートの違いは？

A2. NANDの方が半導体設計時の実装面積に置いて有利。

CMOSによるゲートの実装と半導体の物理的特性に触れることで解説できる

Q2. NANDだけとNORだけロジックゲートの違いは？

A2. NANDの方が半導体設計時の実装面積に置いて有利。

CMOSによるゲートの実装と半導体の物理的特性に触れることで解説できる

Q2. NANDだけとNORだけロジックゲートの違いは？

A2. NANDの方が半導体設計時の実装面積に置いて有利。

CMOSによるゲートの実装と半導体の物理的特性に触れることで解説できる

→ 話が高度なので結論だけ

Q2. NANDだけとNORだけロジックゲートの違いは？

A2. NANDの方が半導体設計時の実装面積に置いて有利。

CMOSによるゲートの実装と半導体の物理的特性に触れることで解説できる

→ 話が高度なので結論だけ

実装面積が NORは14、NANDは10 なので、NANDの方が30%程小さい

今日やること

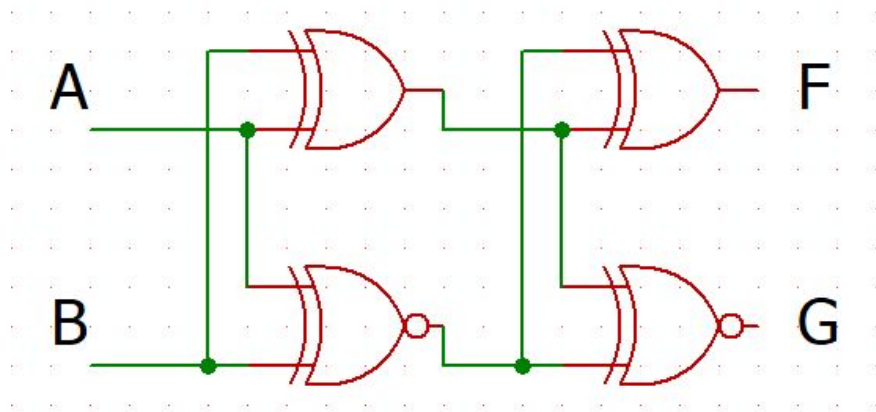
- 告知
- これも立派な計算機
- NOTがあれば何でもできる
- 今日からお前も計算機
- 前回の質問回答
- おまけ

おまけ

Q: XOR、XNORでTrue/False作れる?

A: 作れる

証明(計算)はやってみよう



宣伝

みんなでプログラミングの勉強ができる集会

URC OSS集会

隔週木曜日22時～

```
token = x_token[0]
movie_path = f"movie/{token}.mp4"
temp_movie_path = f"movie/{token}_temp.mp4"
image_path = f"image/{token}.png"
with open(image_path, "rb") as image_file:
    image_data = image_file.read()
MovieMaker.image_to_movie(image_data, temp_movie_path)
name(temp_movie_path, f"movie/{token}.mp4")
["message", "success", "url", "download_url"]

get("/api/create_github_movie/")
create_github_movie(str(movie_path), width=1280, height=int=720,
                    screenshot=True) -> dict:
{
  "url": "https://github.com/noricha/urc-oss",
  "download_url": "https://github.com/noricha/urc-oss",
  "message": "Screenshot taken successfully and take a screenshot."
}

param url: URL to take screenshot
param targets: list of targets to take screenshot
param width: Browser width
param height: Browser height
param limit_height: Max scroll height
param scroll_each:

:param catch: if catch is true, check saved movie is suitable.
:return: Github repository URL
"""
targets = targets.split(" ")
if len(url) == 0:
    raise HTTPException(status_code=400, detail="URL is empty")
```

進捗報告LT, サイトのデバッグ,
コード公開, コードレビューなど

参加方法 「のりちゃん(Noricha)」にフレンド申請&Join

@noricha_vr

個人開発集会

EXPLORE NEW WORLD

完
全
人
に
開
発
解
した

隔週 木曜日 21時～

個人開発やる！ やってる！ やってみたい！
みんな、個人開発についておしゃべりしよう！

ワークショップ

新しい出会い

ためになるLT

参加方法 「のりちゃん (Noricha)」に
フレンド申請&Join

次回は今週の6/22! CS集会と同じ週にやるよ!