

四則演算

- Verilogでは四則演算(+,-,*,/)は普通に書けば、加算、減算、乗算、除算を実行します。しかしソフトウェアと違って、加算器、減算器、乗算器、除算器を生成することに注意してください。除算器などは非常に大量のハードウェアを必要とします。乗算器、除算器はどのような論理合成系でハードウェアを生成するかを良く考える必要があります。

```
assign x = a+b;
```

```
assign y=a*b;
```

論理演算(ANDとOR)

- 論理積(AND) Verilog演算子 &
 - $0&0=0, 1&0=0, 0&1=0, 1&1=1$
 - 多桁の場合、対応するビット間の演算となる
 - 例) $1011 \& 1101 = 1001$
 - 1を検出するマスク操作によく用いる
- 論理和(OR) Verilog演算子 |
 - $0|0=0, 1|0=1, 0|1=1, 1|1=1$
 - 多桁の場合、対応するビット間の演算となる
 - 例) $1001 | 1101 = 1101$

論理演算(NOTとEX-OR)

- 反転(NOT) Verilog演算子 \sim
 - $\sim 0=1, \sim 1=0$
 - 多桁の場合、各ビットを反転する
 - 例) $\sim 1011 = 0100$
 - 1項演算子
- 排他的論理和(EX-OR) Verilog演算子 \wedge
 - $0^0=0, 1^0=1, 0^1=1, 1^1=0$
 - 多桁の場合、対応するビット間の演算となる
 - 例) $1001 \wedge 1101 = 0100$
 - 一致、反一致の判定に使う

シフト(論理シフト)

- 左シフト(Shift Left Logical) Verilog演算子 <<
 - 指定ビット数分左にずらす 2倍、4倍、8倍、、、
 - ずれた分、右(LSB:Least Significant Bit)には0を詰める
 - $11101010 \ll 1 = 11010100$
 - $11101010 \ll 5 = 01000000$
- 右シフト(Shift Right Logical) Verilog演算子 >>
 - 指定ビット数分右にずらす $\frac{1}{2}$ 、 $\frac{1}{4}$ 、 $\frac{1}{8}$ 、、、
 - ずれた分、左(MSB:Most Significant Bit)には0を詰める
 - $11101010 \gg 1 = 01110101$
 - $11101010 \gg 5 = 00000111$
- コンピュータの演算には他にも算術シフト、ローテーションなどありますが、Verilogの演算子としては定義されていません。

比較演算子

- 成立すれば1、そうでなければ0を返す
- 大小比較:< <= > >=
- 等号: == != === !==
 - == !=は、x(不定)、z(ハイインピーダンス)が入力にあれば結果はxやzになる
 - === !==は、x、zを含めて比較する
 - この授業で== !=のみを利用する

リダクション演算

- 論理演算子をバスの前に書くとリダクション演算子となる
- 全ビットを演算し、結果は1か0の1ビットの値になる

A=4'b1001ならば

AND &A=0

OR |A=1

NAND ~&A=1

NOR ~|A=0

演算子の優先順位

| | |
|---------|---------------|
| 論理否定 | !(条件に対する否定) ~ |
| 乗除算 | * / % |
| 加減算 | + - |
| シフト演算 | << >> |
| 比較演算 | < > <= >= |
| 等号 | == != === !== |
| 論理積 | & |
| 排他的論理和 | ^ |
| 論理和 | |
| 論理積(条件) | && |
| 論理和(条件) | |
| 条件 | ? : |