2012年度
計算機システム演習 第9回

計算機システム TA 福田圭祐

本日の内容 (Outline)

- ALUの作成
  - 1ビット加算器の作成
  - 32ビット加算器の作成
  - 1ビットALU
  - 32ビットALU
    - ver.1: 論理積、論理和、加算
    - ver.2: ver.1 + 減算
    - ver.3: ver.2 + 比較演算(slt)
    - ver.4: ver.3 + 等号演算

課題1

- 32ビットALU（ver. 4）を作成せよ
  - 以下のクラスを作る必要がある
    - MUX、MUX4、ALU、ALU_msb、ALU32クラスを定義
  - ALU32Driver クラスを作り、全ての演算についてテストすること

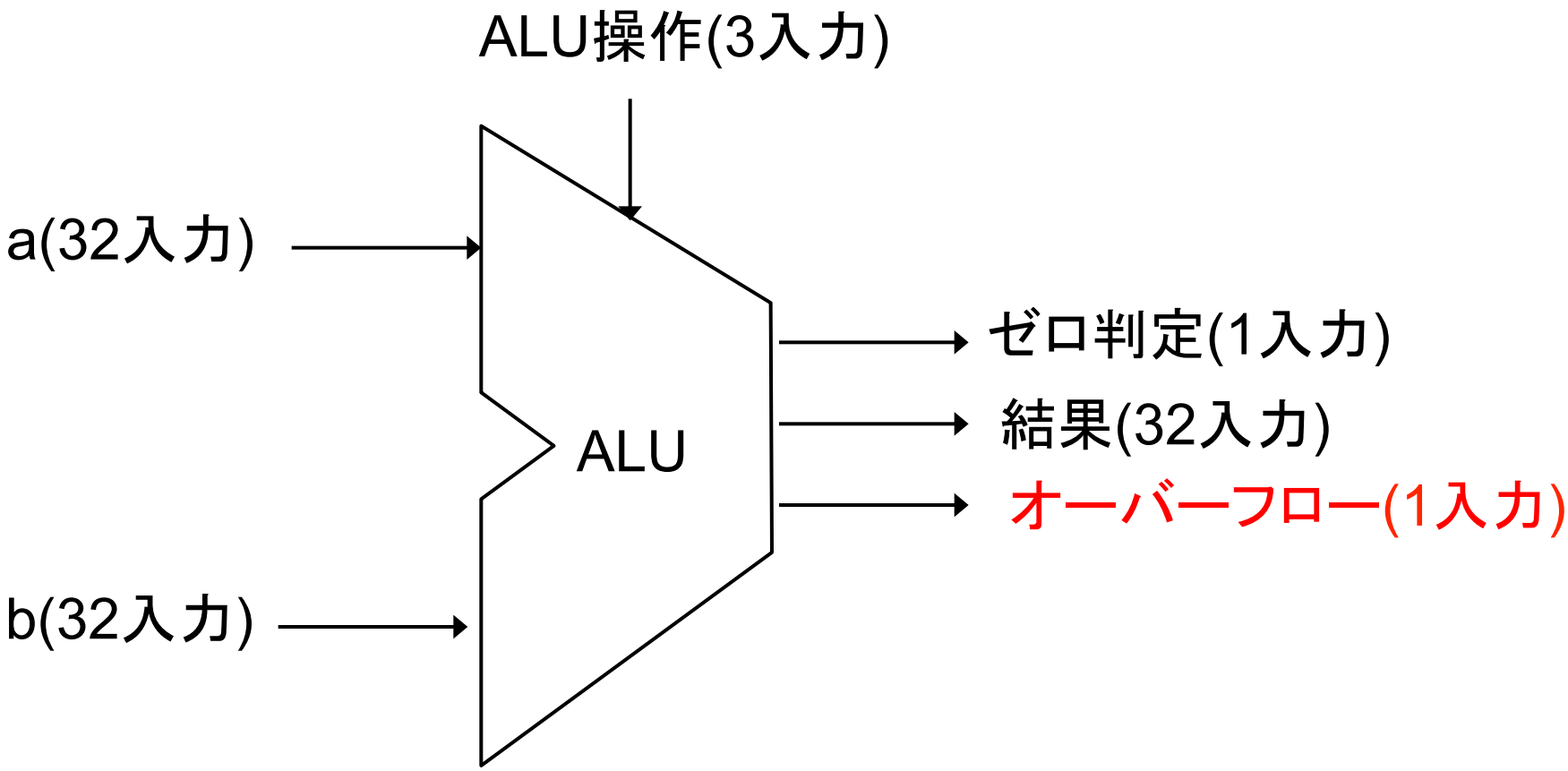
課題2（オプション）

- 32ビットALUにオーバフローを判定する回路を追加せよ
  - MSB 用の ALU に追加すればよい
    - オーバフローをおこすと 1 を出力
    - オーバフローは加減算を行った時に以下の条件で起こる
      - MSB を見れば a, b, s の符号が分かる
      - もちろん符合の判別にifは使用不可

演算	a	b	オーバフローを起こした時の演算結果s
a+b	>=0	>=0	<0
a+b	<0	<0	>=0
a-b	>=0	<0	<0
a-b	<0	>=0	>=0

binvert	a (msb)	b (msb)	s (msb)
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

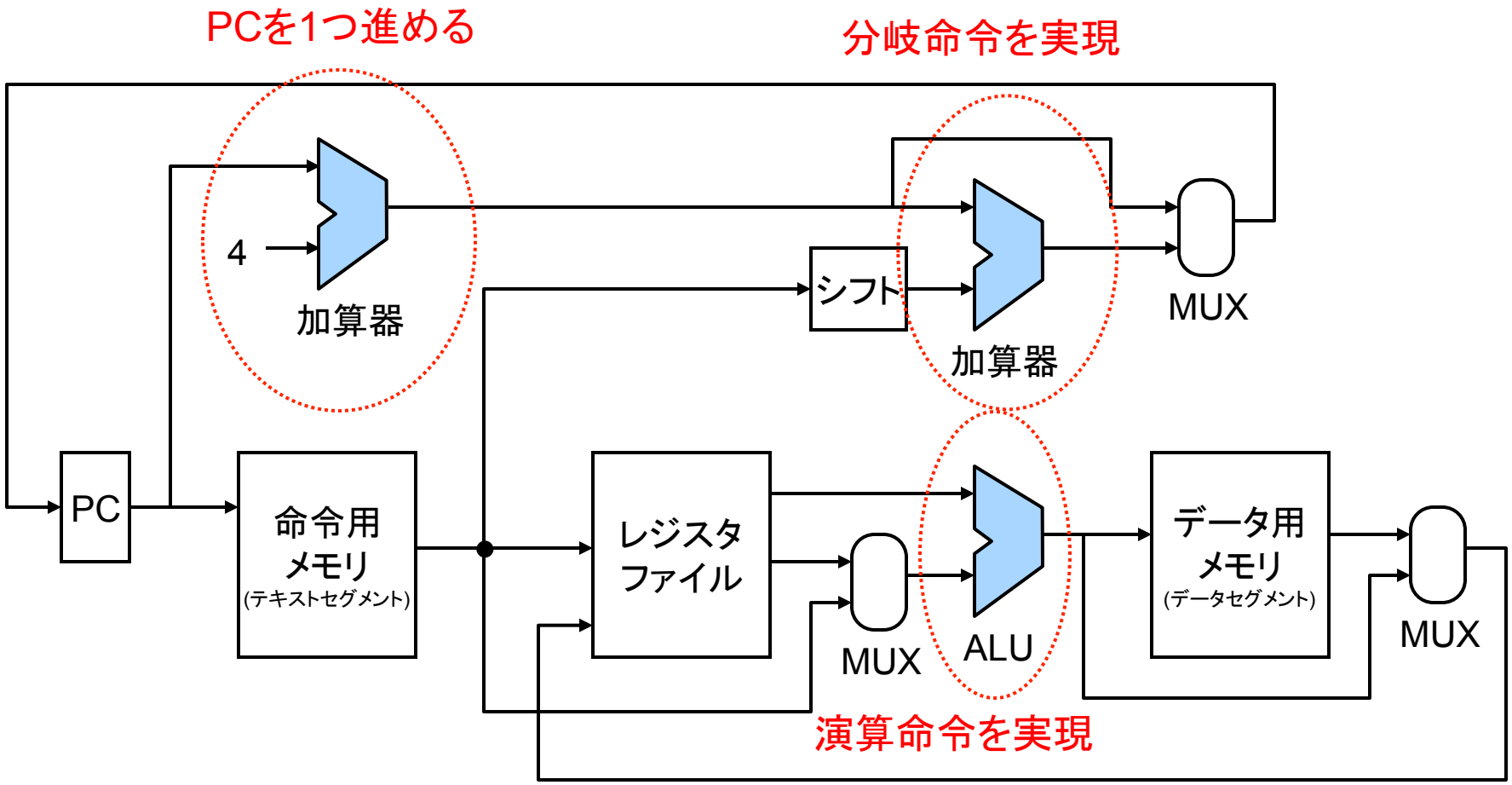
32ビット ALU完成（Overflow）



課題３（オプション）

- 6月18日 に発表される Top500 2012 June を調査し，以下の点についてまとめよ
  - １位になったサイトの，国・機関・性能についてまとめよ．性能には，RPeak・Rmaxの２種類が存在することに注意し，Pflops（ペタフロップス）に換算せよ
  - １０位までを集計し，国別にサイト数をまとめよ．日本は１０位以内に何台はいっているか？

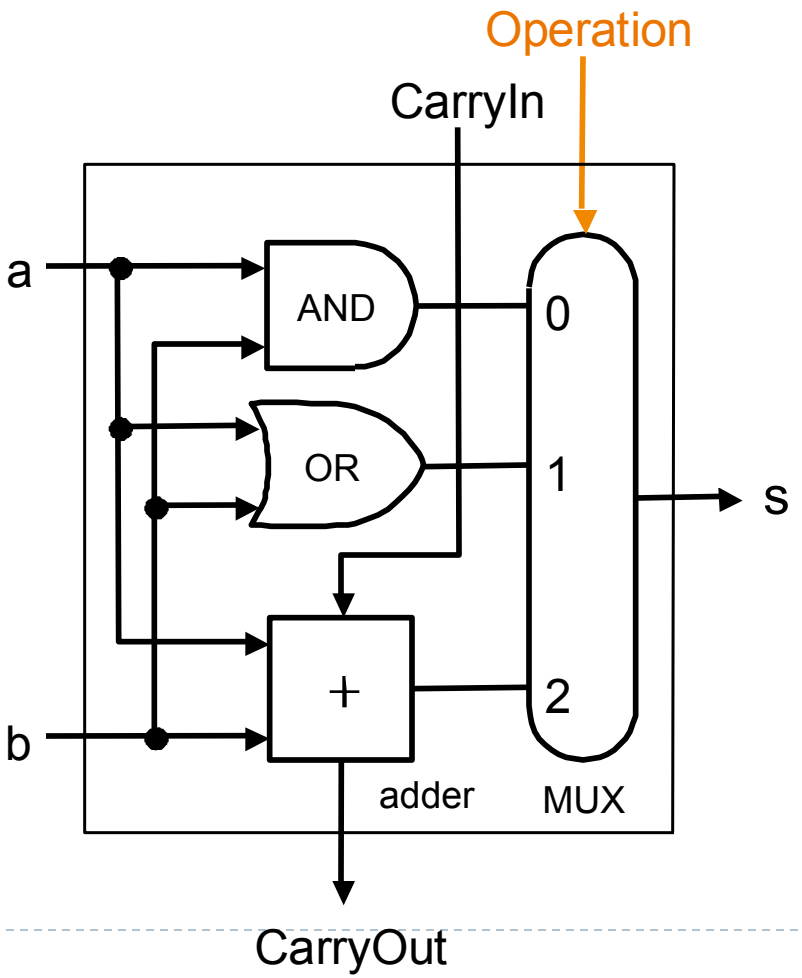
MIPSシミュレータの概略図



1ビットALU（ver. 1）

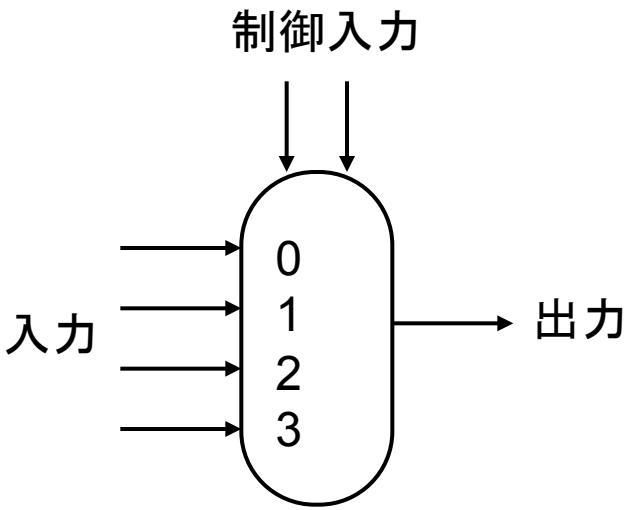
- まず、加算、論理積、論理和をサポート
- マルチプレクサ(MUX)でどの結果を出力するかを決定
  - ANDGate
  - ORGate
  - Adder

Operation	演算
00	AND
01	OR
10	加算
11	-



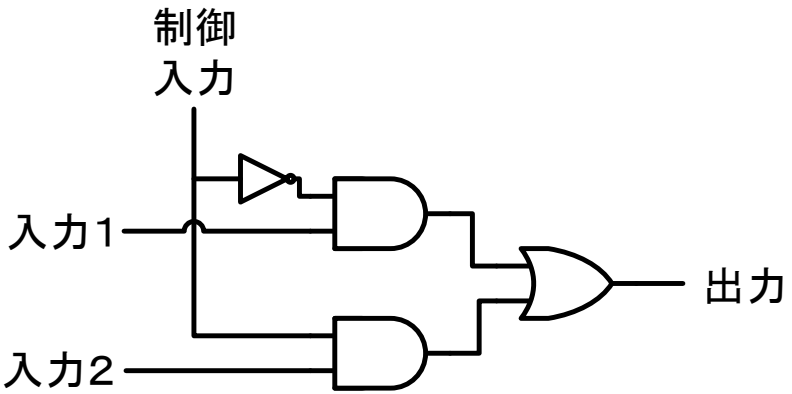
マルチプレクサ(MUX)

- 制御信号に従って、複数の入力から1つの出力を得る
  - 制御信号が表す二進数に対応する入力が選ばれる
  - 制御入力: n ビット => 2^n通りの選択

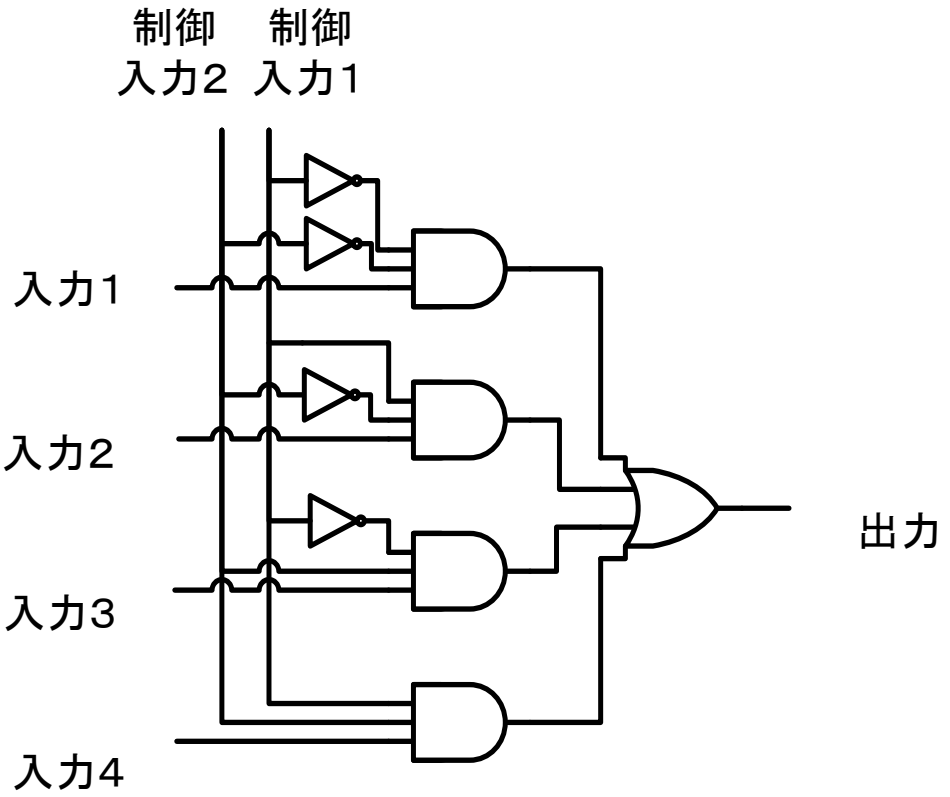


制御信号	選ばれる入力
00	0 番
01	1 番
10	2 番
11	3 番

マルチプレクサの回路図



2入力



4入力

MUX クラス, MUX4 クラス

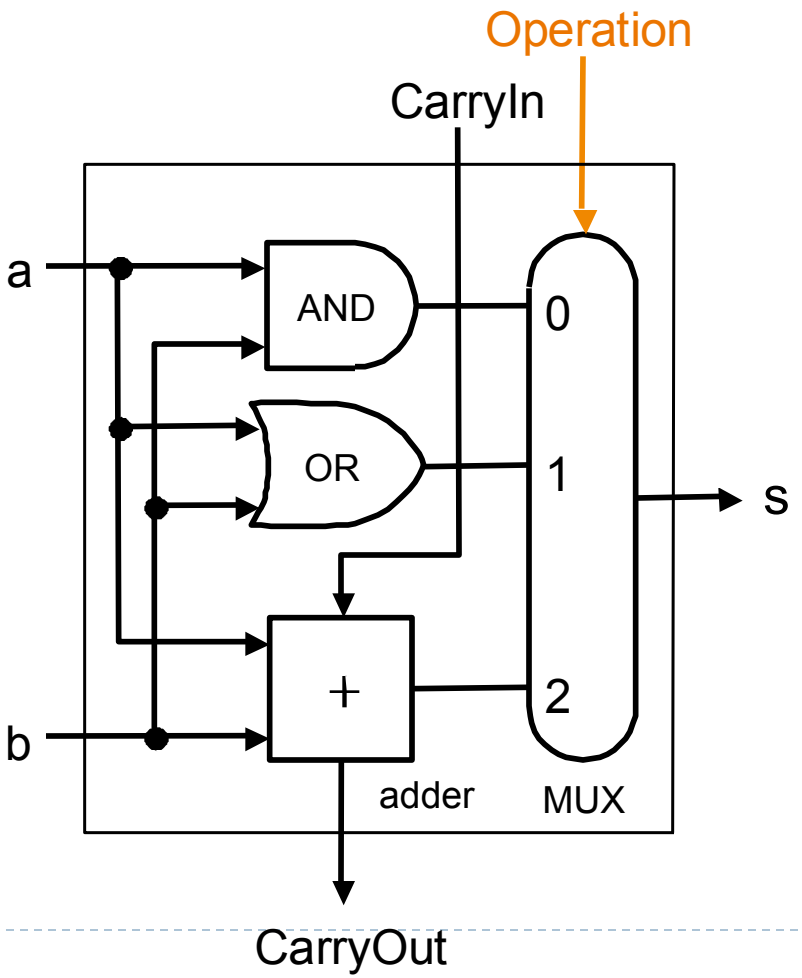
```
public class MUX {
  public MUX(Path ctl,                  // 制御入力
             Path in1, Path in2,        // 入力
             Path out1) {               // 出力
    :
  }
  public void run() { ... }
}
```

```
public class MUX4 {
  public MUX4(Path[] ctls,   // 制御入力
              Path[] ins,    // 入力
              Path out1) {   // 出力
    :
  }
  public void run() { ... }
}
```

1ビットALU（ver. 1）　（再）

- まず、加算、論理積、論理和をサポート
- マルチプレクサ(MUX)でどの結果を出力するかを決定
  - ANDGate
  - ORGate
  - Adder

Operation	演算
00	AND
01	OR
10	加算
11	-



1ビットALU （ver. 1）クラス

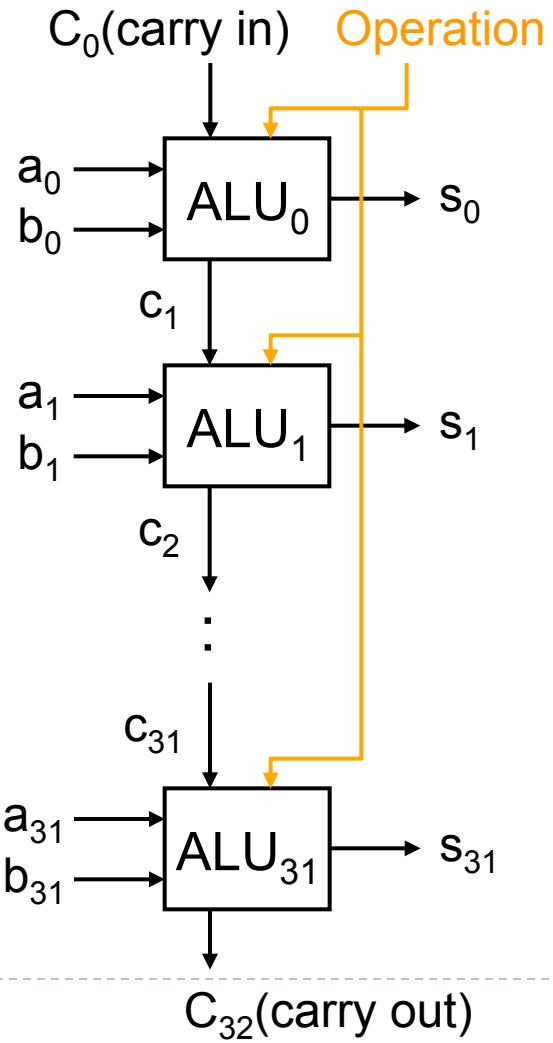
```
public class ALU {
  public ALU(Path[] op,                              // 制御入力
             Path a, Path b, Path carryIn,           // 入力
             Path s, Path carryOut) {                // 出力
    Path[] inner = new Path[4];

    // ANDGate, ORGate, FAを作成

    mux1 = new MUX4(op, inner, s);
  }
  public void run() {
    // 全ての演算を計算して、最後にマルチプレクサで選択する
    // 演算の種類に応じてif文で分岐してはならない
    // つまり、回路の全ての要素についてrun()を呼ぶ必要がある
  }
}
```

32ビットALU（ver. 1）

- 1ビットALU を 32 個つなぐ
  - Ripple Carry Adder で実装



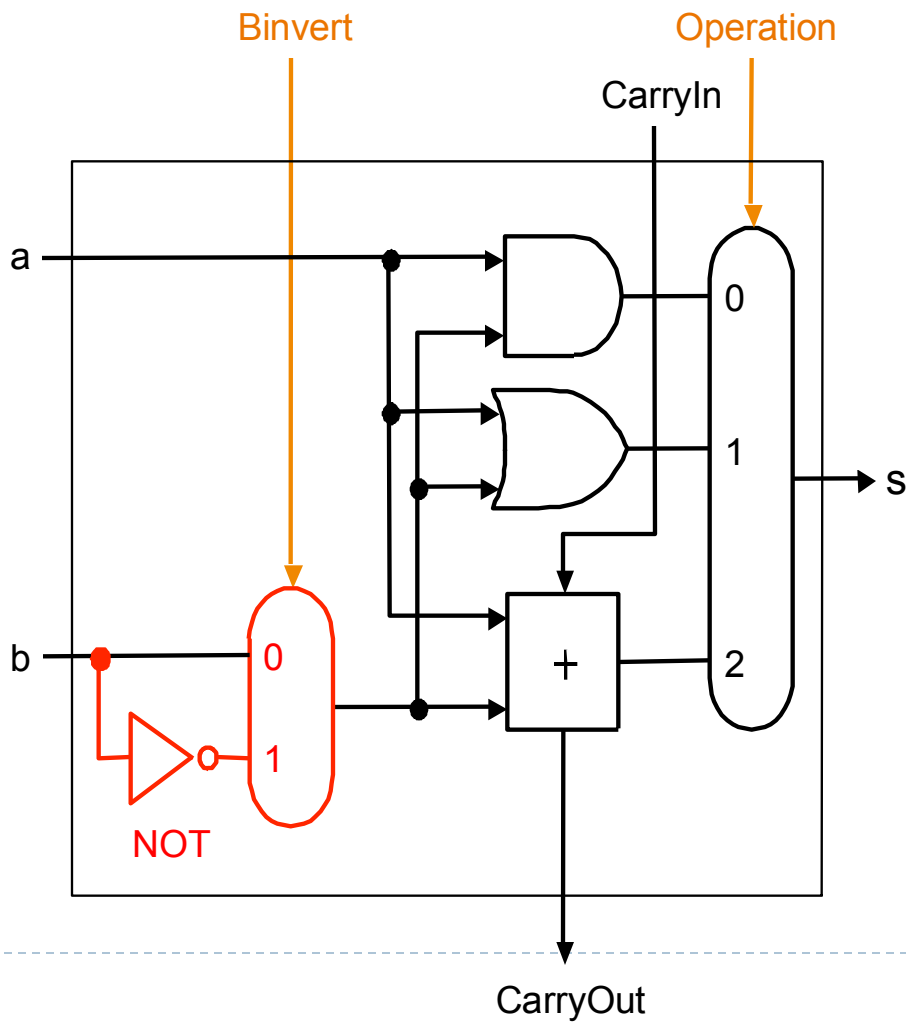
32ビットALU（ver. 1）クラス

```
public class ALU32 {
  ALU[] alu = new ALU[32];
  public ALU32(Path[] op,                          // 制御入力
               Bus a, Bus b, Path carryIn,         // 入力
               Bus s, Path carryOut) {             // 出力
    :
  }
  public void run() {
    : // 32bit ALUを順番に実行
  }
}
```

1ビットALU (ver.2)

- 減算をサポート
- Binvert によるマルチプレクサを追加
- 制御入力と演算の関係は以下の通り

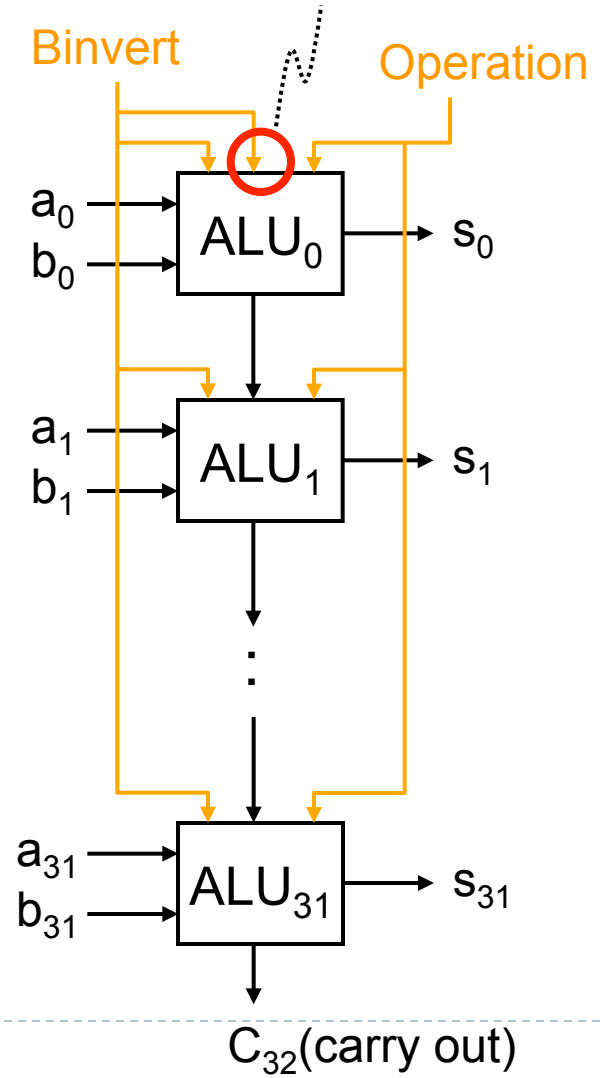
Binvert	Operation	演算
0	00	AND
0	01	OR
0	10	加算
1	10	減算



32ビットALU（ver. 2）

- 減算をサポート
  - 2 の補数で計算
    - 入力 b のビットを反転
      - Binvert が 1 の時
    - a と b を加算
    - 同時に 1 を加算
      - Binvert を CarryIn にすることで実現
  - 4ビットでの例
    - 0010 - 0001
      = 0010 + 1110 + 1
      = 0001

2の補数
(= ビット反転して1足す)



32ビットALU（ver. 2）クラス

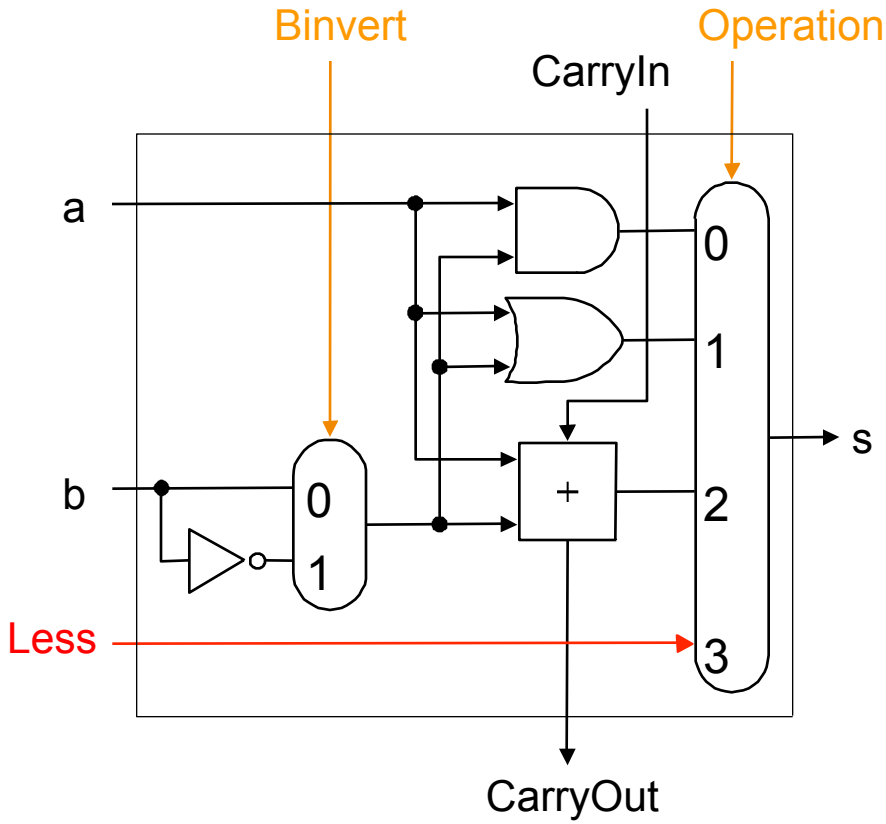
```
public class ALU32 {
  ALU[] alu = new ALU[32];
  public ALU32(Path binvert, Path[] op,     // 制御入力
               Bus a, Bus b, Path carryIn   // 入力
               Bus s, Path carryOut) {      // 出力
    :
  }
  public void run() {
    :
  }
}
```

注意：最下位ビットのCaryyInにはbinvertをセット

1ビットALU （ver. 3）

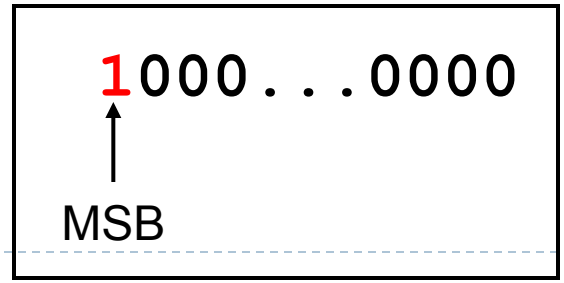
- sltのサポート
- 入力 Less を追加
  - MUX で選択されると Less の値をそのまま出力

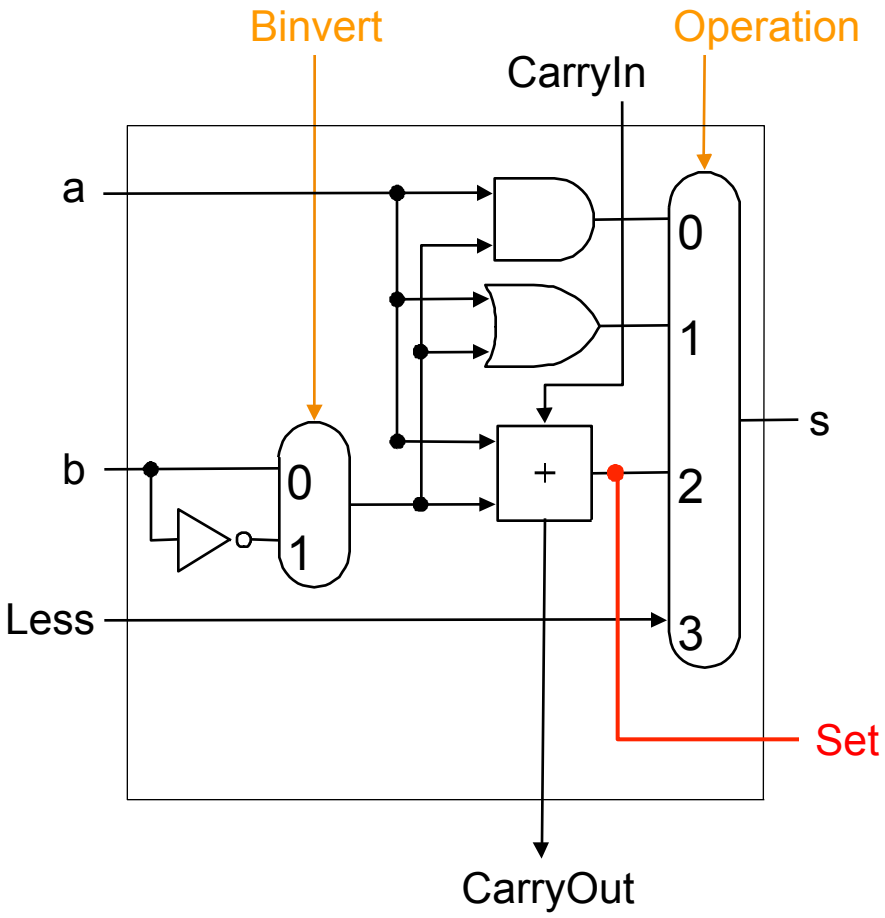
Binvert	Operation	演算
0	00	AND
0	01	OR
0	10	加算
1	10	減算
1	11	slt



1ビットALU （ver. 3, MSB用）

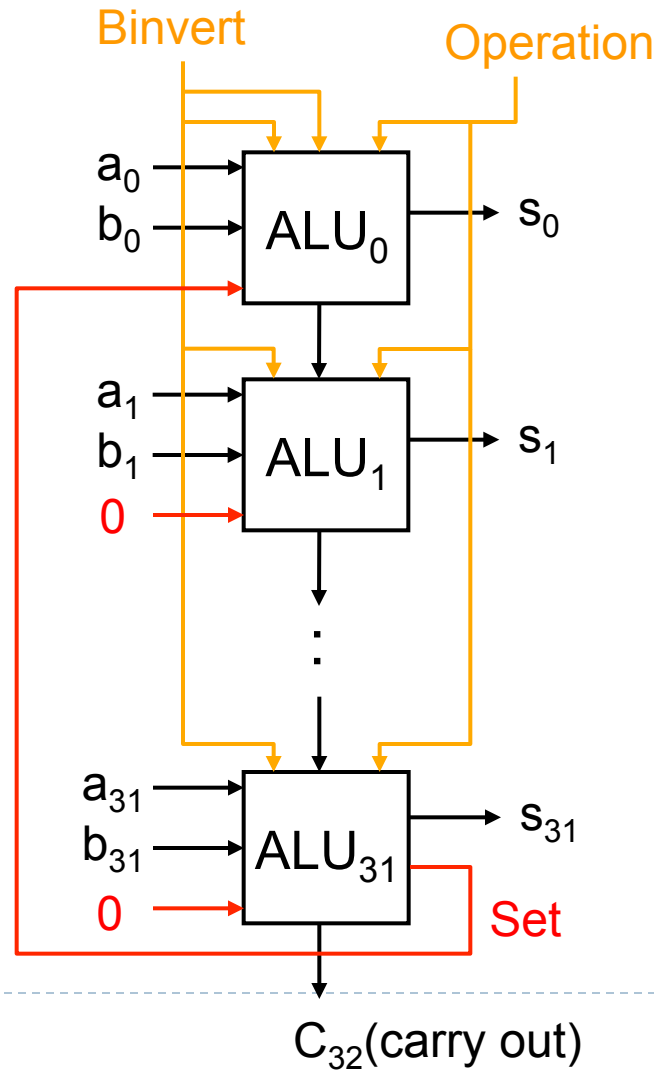
- 最上位ビット（MSB）の ALU は Set も出力する
  - MSB は 32 ビットの数値の正負を表す（2 の補数）
    - 0 なら正
    - 1 なら負
  - Set はa – b の結果が負になると 1 になる





32ビットALU（ver. 3）

- 比較演算（slt）をサポート
  - a < b の時に 1 を出力
    - a – b を計算
    - 結果が負なら Set の値は 1
    - MUX4で3 つ目の入力を出力する
  - ALU を 2 回通る（2回実行）
    - 減算を実行
      - Set の値を計算するため
    - 3 つ目の入力を出力
      - つまり、Set の値



1ビットALU（ver. 3）クラス

```
public class ALU {
  public ALU(Path binvert, Path[] op,     // 制御入力
                  Path a, Path b, Path less,  // 入力
                  Path carryIn,
                  Path s, Path carryOut,      // 出力
                  ) {
    :
  }
  public void run() {
    :
  }
}
```

1ビットALU （ver. 3, MSB用）クラス

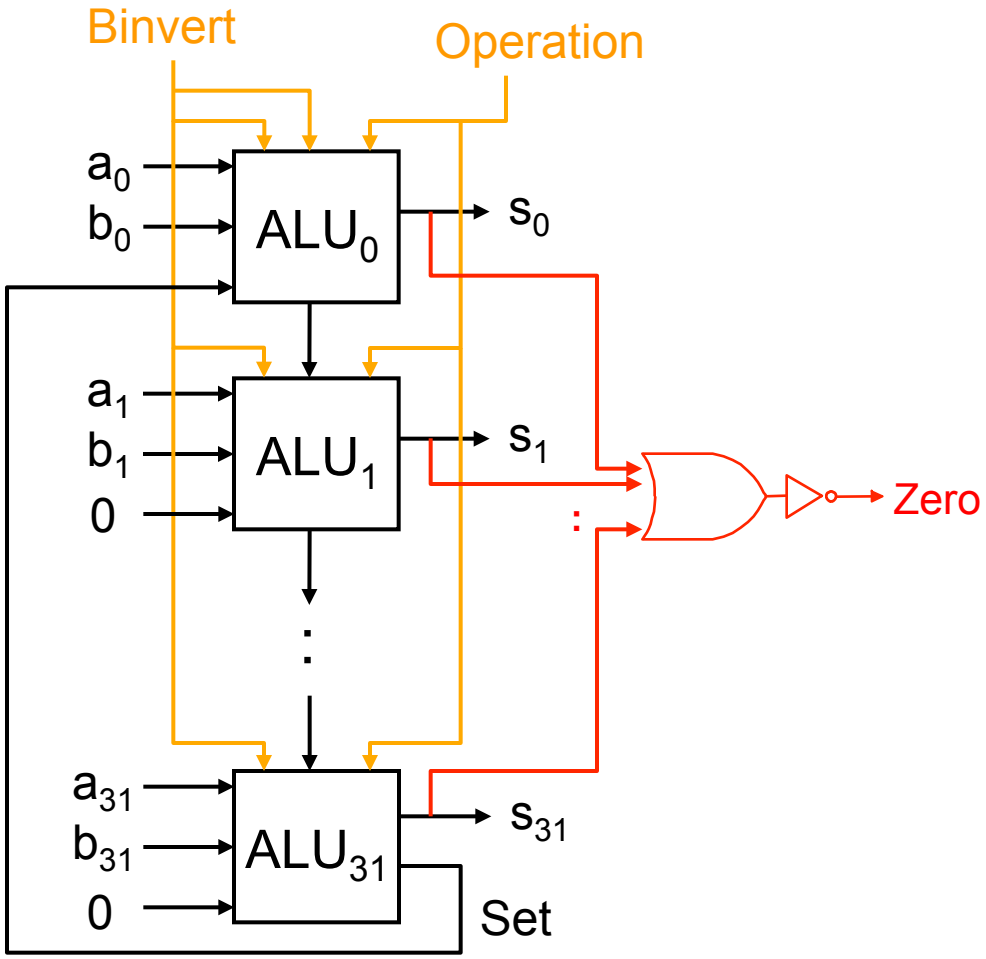
```
public class ALU_msb {
  public ALU_msb(Path binvert, Path[] op,     // 制御入力
                 Path a, Path b, Path less,   // 入力
                 Path carryIn,
                 Path s, Path carryOut,       // 出力
                 Path set) {
    :
  }
  public void run() {
    :
  }
}
```

32ビットALU（ver. 3）クラス

```
public class ALU32 {
  ALU[] alu = new ALU[31];
  ALU_msb alu_msb;
  public ALU32(Path binvert, Path[] op,   // 制御入力
               Bus a, Bus b,              // 入力
               Bus s, Path carryOut) {    // 出力
    :
  }
  public void run() {
    // sltの場合を考慮して、ALUを2回実行する必要がある
    //  1回目はSetの計算
    //  2回目はLessの値の出力
  }
}
```

32ビットALU（ver. 4）

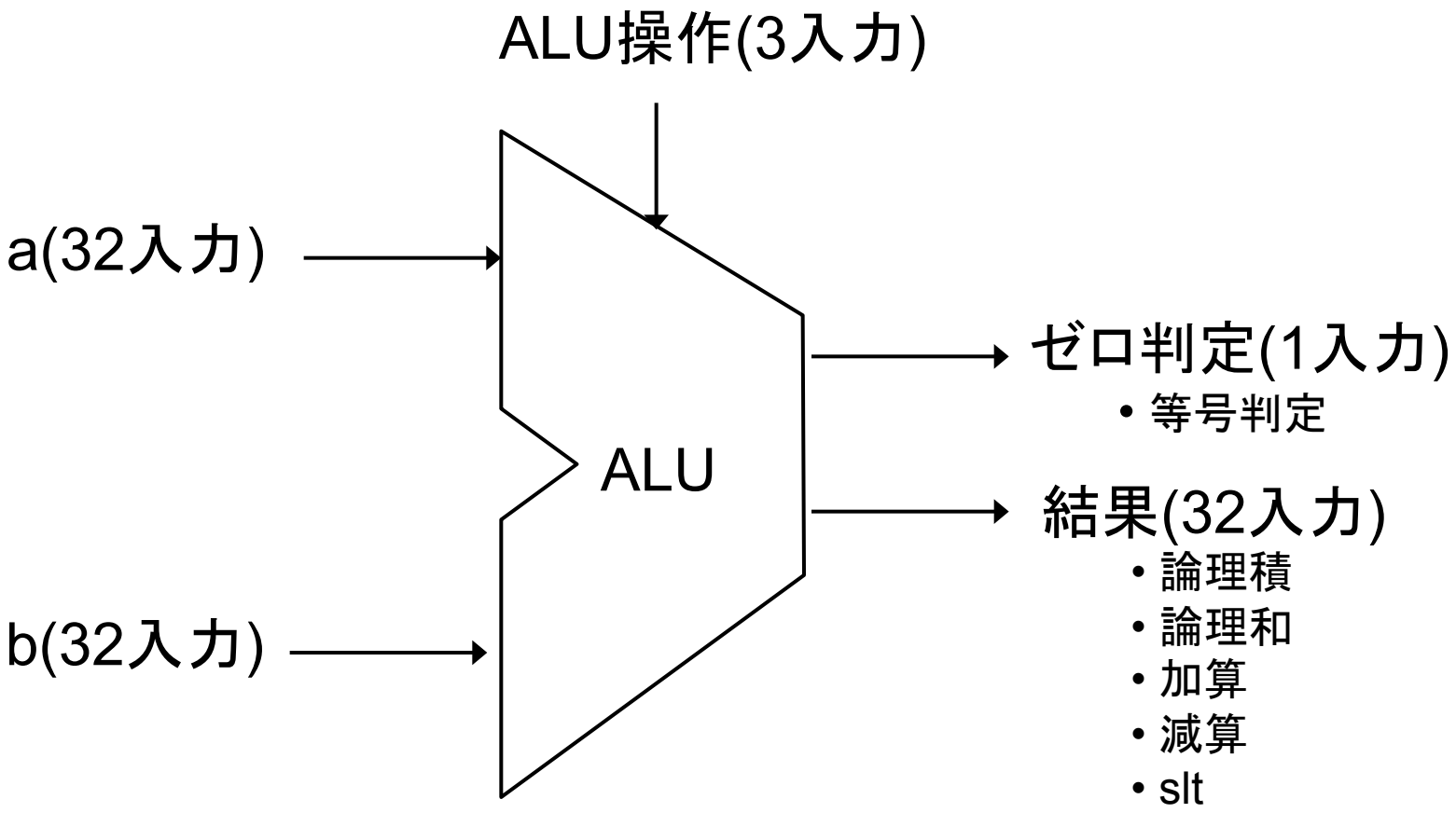
- 等号演算をサポート
  - 条件分岐等で使用
  - 減算演算結果が 0 なら Zero を 1 にする
    - すべてのビットが 0 になった時
  - 制御入力
    - Binvert: 1
    - Operation: 10



32ビットALU（ver. 4）クラス

```
public class ALU32 {
  // 今後のためにBinvertとOperationはまとめる
  // carryOutは使わないので省略
  public ALU32(Path[] ops,            // 制御入力
               Bus a, Bus b,          // 入力
               Bus s, Path zero) {    // 出力
    Path binvert = ops[2];
    Path[] op = new Path[2];
    op[1] = ops[1];
    op[0] = ops[0];
      :
    new ALU(binvert, op, ...);
  }
  public void run() {
    :
  }
}
```

32ビット ALU完成


ALU操作(3入力)
a(32入力)
b(32入力)
ALU
ゼロ判定(1入力)
• 等号判定
結果(32入力)
• 論理積
• 論理和
• 加算
• 減算
• slt

課題

課題1 （再掲）

- 32ビットALU（ver. 4）を作成せよ
  - 以下のクラスを作る必要がある
    - MUX、MUX4、ALU、ALU_msb、ALU32クラスを定義
  - ALU32Driver クラスを作り、全ての演算についてテストすること

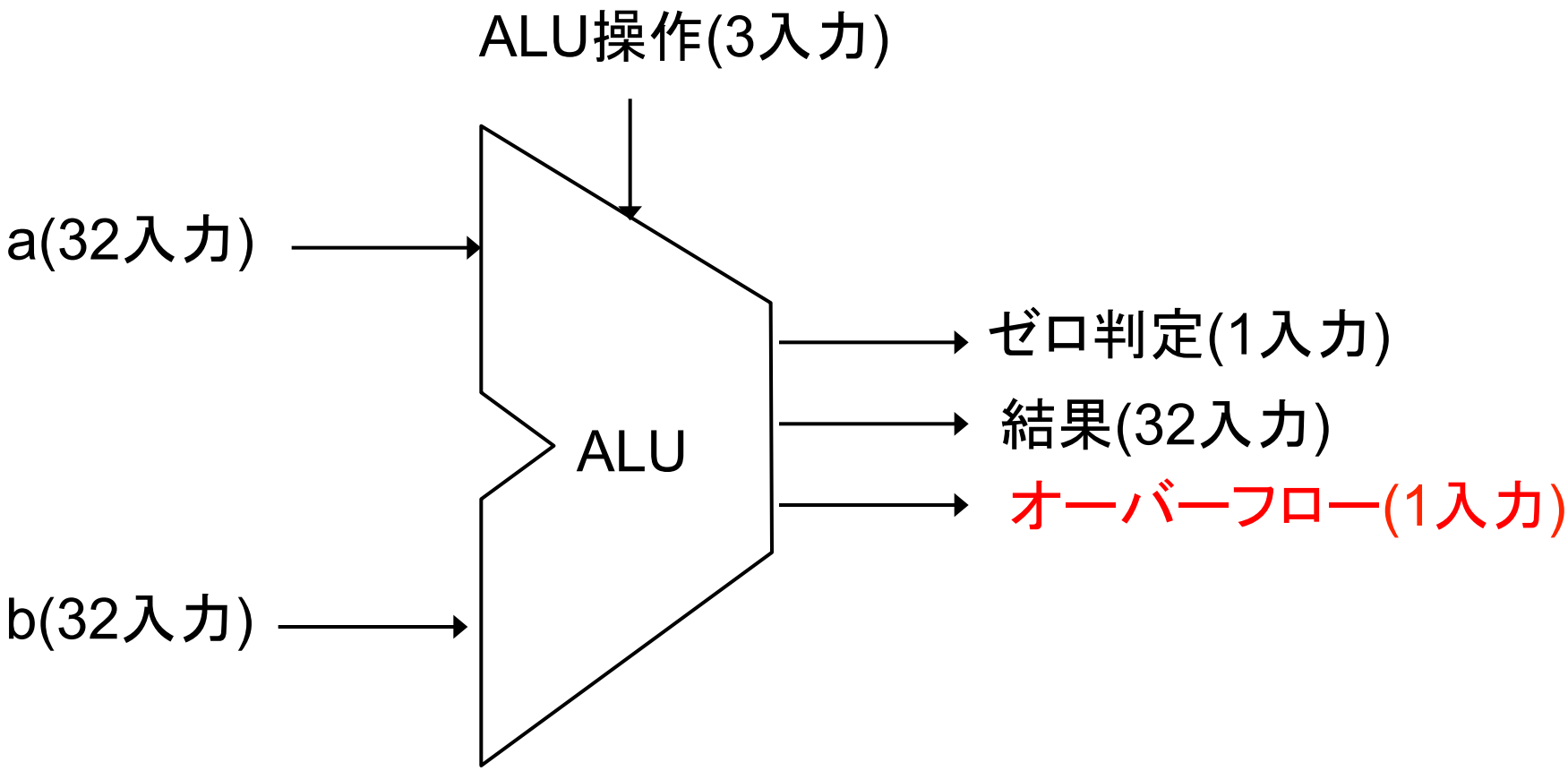
課題2（オプション）（再掲）

- 32ビットALUにオーバフローを判定する回路を追加せよ
  - MSB 用の ALU に追加すればよい
    - オーバフローをおこすと 1 を出力
    - オーバフローは加減算を行った時に以下の条件で起こる
      - MSB を見れば a, b, s の符号が分かる
      - もちろん符合の判別にifは使用不可

演算	a	b	オーバフローを起こした時の演算結果s
a+b	>=0	>=0	<0
a+b	<0	<0	>=0
a-b	>=0	<0	<0
a-b	<0	>=0	>=0

binvert	a (msb)	b (msb)	s (msb)
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

32ビット ALU完成（Overflow）



課題３（オプション）（再掲）

- 6月18日 に発表される Top500 2012 June を調査し，以下の点についてまとめよ
  - １位になったサイトの，国・機関・性能についてまとめよ．性能には，RPeak・Rmaxの２種類が存在することに注意し，Pflops（ペタフロップス）に換算せよ
  - １０位までを集計し，国別にサイト数をまとめよ．日本は１０位以内に何台はいっているか？

課題提出

- 〆切：7/6 (Fri) 23:59
- 提出物：以下のファイルを圧縮したもの
  - ドキュメント(pdf,plain txt,wordなんでも可)
    - テスト結果　(テスト内容とその結果)
      - 注意：作成したプログラムは今後も使用するため、十分にテストすること
    - 感想等
  - プログラムソース一式 (ソースコードのみ)
    - 必ず…Driver.javaクラスも含める
- 提出方法：Webから提出