always文

always文は@以下の条件が成り立つときに常に実行される
posedge 立ち上がり　negedge 立ち上がり
or, and はここだけで使う特殊な条件指定論理

決まった形式以外は使わない！

```
reg accum;
always @(posedge clk or negedge rst_n)
 begin
   if(!rst_n) accum <= 16'b0;
   else accum <= alu_y;
 end
```

レジスタに対する値の書き込みは＜＝を使ってalways文の中で行う

always文中ではif文やcase文が使えるなぜか？

レジスタに対する代入だから→プログラム言語の変数と同じで代入されない場合の値が決まっている

非同期リセットと同期リセット

rst_nがLの時は常にリセット→非同期リセット

```
always @(posedge clk or negedge rst_n)
 begin
   if(!rst_n) accum <= 16'b0;
   else accum <= alu_y;
 end
```

rst_nがLでクロックが立ち上がりの時にリセット→同期リセット

```
always @(posedge clk )
 begin
   if(!rst_n) accum <= 16'b0;
   else accum <= alu_y;
 end
```

最近FPGAなどでは非同期リセットが主流。このためこの授業でも非同期リセットを使う

always文の注意

- 原則としてあるレジスタに対する値の書き込みは個別のalways文で行うことをお勧めします。つまり、一つのalways文中で複数のレジスタに対して値の書き込みを行うことは避けた方が無難です。
- 非常に関連が深く、一括して扱った方がわかりやすい場合のみ、複数のレジスタに同じalways文で書き込んでください。