

# if文

if文は、C言語のif文に似ています。

```
if(条件)
```

```
begin 文,文, ...end
```

```
else
```

begin 文,文,..end で、条件が成立すれば最初のbegin内の文が成立しなければelse以降の文が実行されます

```
reg accum;
```

```
always @(posedge clk or negedge rst_n)
```

```
begin
```

```
if(!rst_n) accum <= 16'b0;
```

```
else accum <= alu_y;
```

```
end
```

# if文の注意

- always文の中だけに使えます。
- reg文で宣言されたレジスタに対する値の書き込みに限定されます。
- C言語同様
  - if(条件1)... else if(条件2)... else if(条件3)  
...else と書いて条件1, 2, 3を順に調べていくことができます。
  - 入れ子に書けますが、この場合きちんとbegin endを付けて文の構造を明解にしてください。

# 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
```

```
always @(posedge clk )
```

```
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

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