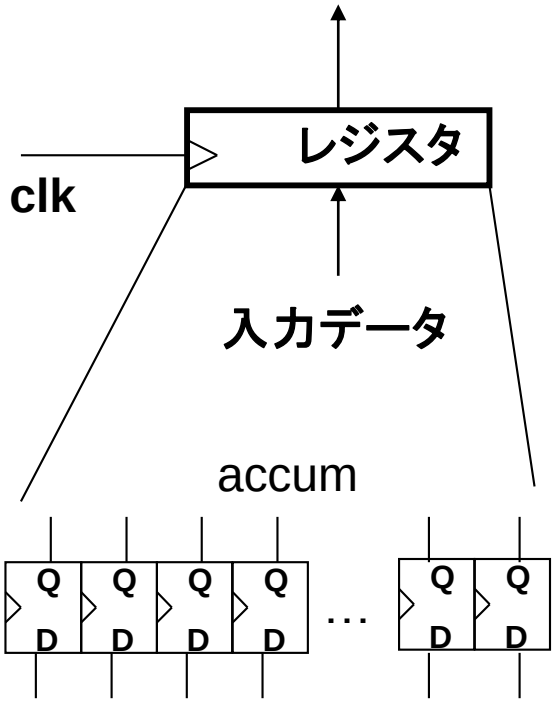
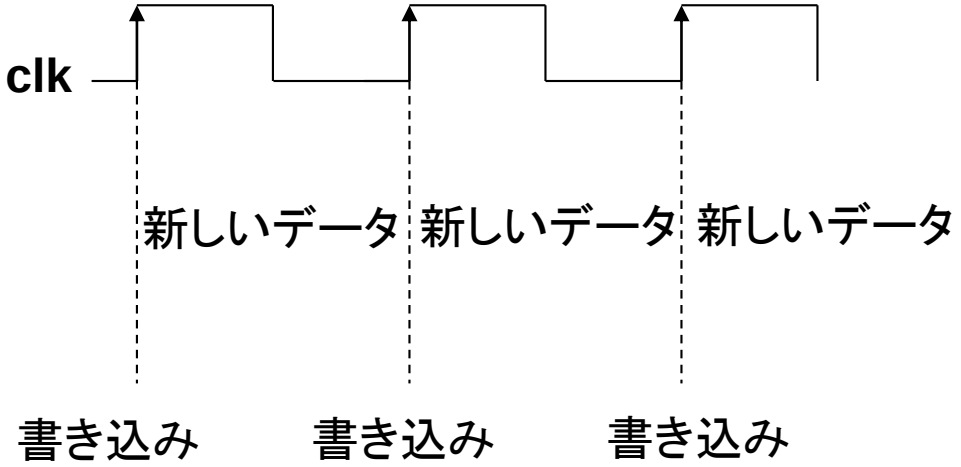
レジスタ

レジスタ＝D.F.Fの集合





clkの立ち上がり（立下り）に同期して書き込む
→CPUの状態はclkに同期して変化する

レジスタのVerilog記述（非同期リセット）

```
reg [15:0] accum;
always @(posedge clk or negedge rst_n)
 begin
   if(!rst_n) accum <= 16'b0;
   else accum <= din;
 end
```

宣言

クロックの立ち上げ同期して書き込み

rst_nが0になると初期化（非同期リセット）

最近は、リセットのタイミングを気にしなくて良い非同期リセットが良く使われる

レジスタのVerilog記述（同期リセット）

```
reg [15:0] accum;
always @(posedge clk)
 begin
   if(!rst_n) accum <= 16'b0;
   else accum <= din;
 end
```

宣言

クロックの立ち上げ同期して書き込み

rst_nが0になり、clkが立ち上がったときにリセットされる（同期リセット）

同期リセットはD-FFがリセット入力を持たなくても実現できる。
しかし、ファンアウトの多いリセット信号がデータ信号と混ざってしまうので
タイミング設計上面倒なことになる場合がある。このため最近あまり用いられない。
一時期利用が奨励されたので、上司のシニア設計者との意見の相違に注意！

レジスタのVerilog記述（イネーブル付き、非同期リセット）

```
reg [15:0] accum;
always @(posedge clk or negedge rst_n)
 begin
   if(!rst_n) accum <= 16'b0;
   else if(we) accum <= din;
 end
```

weが1の時のクロックに同期して値を書き込む

we(write enable)がないと全てのクロックの立ち上がりで値が書き込まれてしまう。
書き込みたい時だけwe=1にする。
最近のRTL設計では、最も良く使われる