

# Chattering Killer Logic in Verilog-HDL

この教材は、情報科学部卒業生の  
作品を利用しています。

Verilogベースの記述ですが、  
XMOSでのサンプルプログラムの説  
明にも言及しています。

# What is Chattering ?

- Chatの語源：友好的な談笑、おしゃべり
- Chatterの意味
  - くだらないおしゃべり（**比較** chat と異なり悪い意味合いに用いられる）.
  - 1 キーキー鳴く声.
  - 2（機械・歯などの）ガチガチ[ガタガタ]する音.

# What is Chattering ? (cont)

- 雑音成分
- 不安定状態
- 信号がON/OFFを繰り返す現象



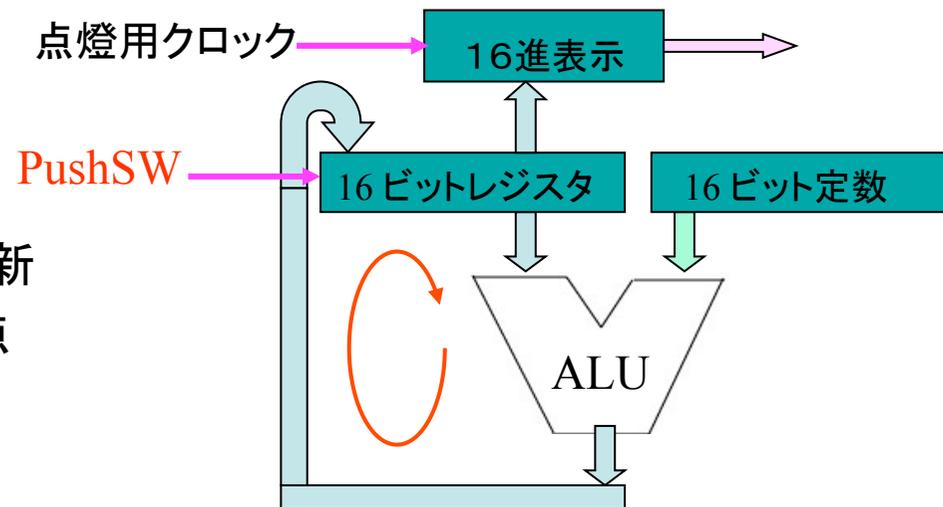
# What causes Chattering?

- リレー接点やスイッチ接点の切り替わり
- チャタリングの継続時間は、数100us～数ms
- スイッチにおけるチャタリング
  - ON: 接点同士が衝突した瞬間には安定せず、ばたつきながら接触状態に落ち着く
  - OFF: 接触圧が徐々に弱まり、接触不良状態になりながらやがて接点が離れる

# Harm of Chattering

- 入力時の誤動作
- プッシュボタンをレジスタのクロックとしてそのまま使った例:

- チャタリングが発生
- 何度も演算を実行し、レジスタを更新
- 回路はチャタリングを複数回の接点動作として捉えてしまう



# Chattering.v

Verilog-HDL

```
// Chattering
// クロックは2KHzを使用
// SW入力が2KHzの間隔でクロックの立ち上がりで
// 2回連続1の状態であった場合、SWがONされたと見なす。
```

```
module chattering(clk,reset,SW,Q);
  input clk, reset, SW;
  output Q;
  reg r_SW, r_Q;
```

```
// チャタリング除去
```

```
always @(posedge clk or negedge reset) begin
```

```
  if(reset == 1'b0) begin
    r_SW <= 1'b0;
    r_Q <= 1'b0;
```

} リセットによる初期化

```
  end else begin
```

```
    if(r_SW == SW) begin
      r_Q <= SW;
```

} 1クロックサイクル前と同じスイッチの状態

```
    end else begin
```

```
      r_SW <= SW;
```

} 1クロックサイクル前と異なるスイッチの状態

```
    end
```

} その状態を記憶

```
  end
```

```
end
```

```
// 出力
```

```
  assign Q = r_Q;
```

```
endmodule
```

# LTCP V1.2

## サンプルプログラム中の処理

- Line 13:

```
#define CHATTERING_R 250000 // 2.5 ms:右用
#define CHATTERING_L 300000 // 3.0 ms:左用
/* 左側だけチャタリングが激しいので、タイマーを長くした */
```

- Line 639

```
p_button_0 := button0;
// ボタン1の処理(右のボタン)
if( button0 != button0_prev ){ // 状態が変化した
    // チャタリングキラー
    if( button0==BTN_ON ){ // ONになった時のみ、チャタリング処理をする。
        chatTimer := timeChat;
        timeChat += CHATTERING_R;
        chatTimer when timerafter(timeChat) := void;
        p_button_0 := button0; // ポートの受信
    }
}
```

# タイミングチャート

