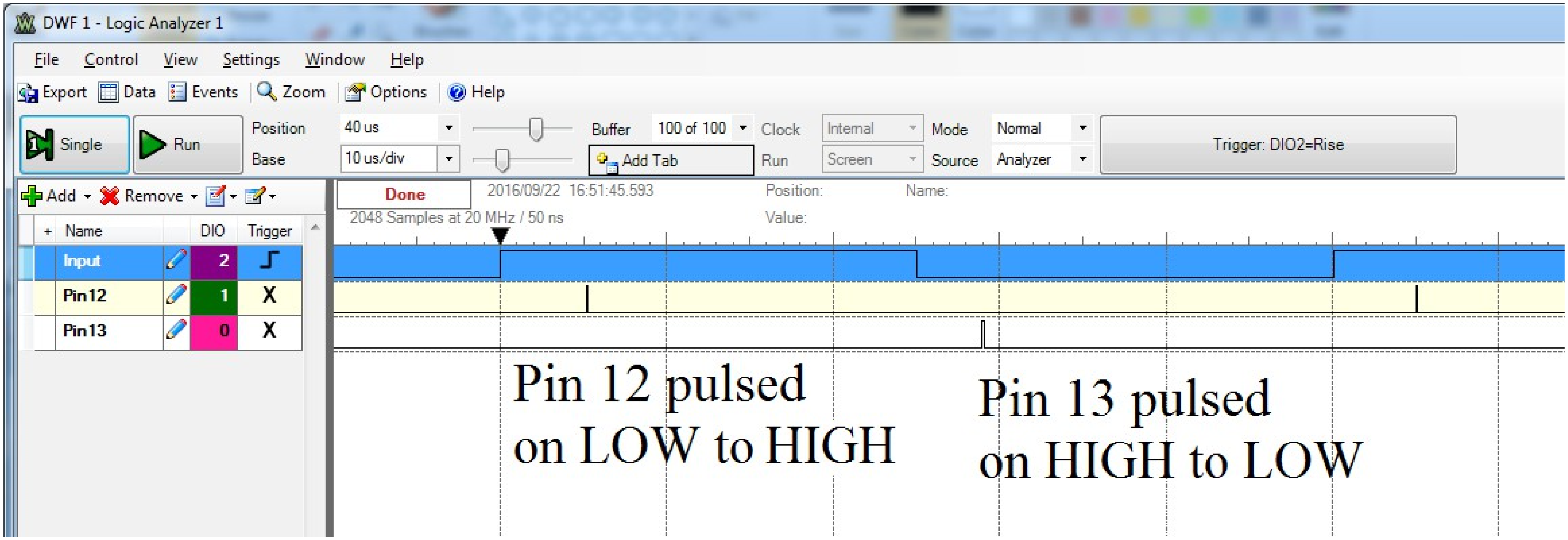
First State Machine

```
// Create a set of states and actions
// for simple state machine.
enum PStates { High, Low };
enum PActions { None, Pulse13, Pulse12 };
// Create state and action for SM
PStates PulseState = High;
PActions PulseAction = None;

// This function will handle the state machine,
// returning an action that is to be done.
PActions PulseNextState(int Input)
{
        // This is the action to be returned,
        PActions ReturnValue = None; // defaults to No action
        // Switch statement for state machine
        switch (PulseState)
        {
          case High: // State indicating input has been high.
              if (Input == LOW) // then once it goes low
              {
                     ReturnValue = Pulse13; // load action into return value.
                     PulseState = Low;      // and change state.
              }
              break;
          case Low: // State indicating input has been low.
              if (Input == HIGH) // then once it goes high
              {
                     ReturnValue = Pulse12; // load action
                     PulseState = High;     // and change state.
              }
              break;
        } // End of State switch

        return ReturnValue;

} // End of PulseNextState

// put your setup code here, to run once:
void setup() {
        DDRB |= 0x30; // Set 13 and 12 to outputs
        bitClear(DDRB, 1); // DDRB &= 0xFD; // Set 9 to input.
        PORTB &= 0x30; // &= 0xCF; // Set 13 and 12 to LOW.
} // End of setup

// put your main code here, to run repeatedly:
void loop() {
         // Check what action is needed based on input on pin 9.
        PulseAction = PulseNextState(bitRead(PINB, 1));
        switch (PulseAction)
        {
          case Pulse13: // Call to pulse pin 13.
              bitSet(PORTB, 5); // PORTB |= 0x20; // Force high
              bitClear(PORTB, 5); // PORTB &=~0x20; // Force low
              break;
          case Pulse12: // Call to pulse pin 12.
              bitSet(PORTB, 4); // PORTB |= 0x10; // Force high
              bitClear(PORTB, 4); // PORTB &=~0x10; // Force low
              break;
          case None:
              break;
        } // End of switch
} // end of loop.
```



DWF 1 - Logic Analyzer 1

Position: 40 us

Base: 10 us/div

Buffer: 100 of 100

Add Tab

Clock: Internal

Run: Screen

Mode: Normal

Source: Analyzer

Trigger: DIO2=Rise

Done

2016/09/22 16:51:45.593

2048 Samples at 20 MHz / 50 ns

Position:

Value:

Name:

+ Name	DIO	Trigger
Input	2	⎍
Pin12	1	X
Pin13	0	X

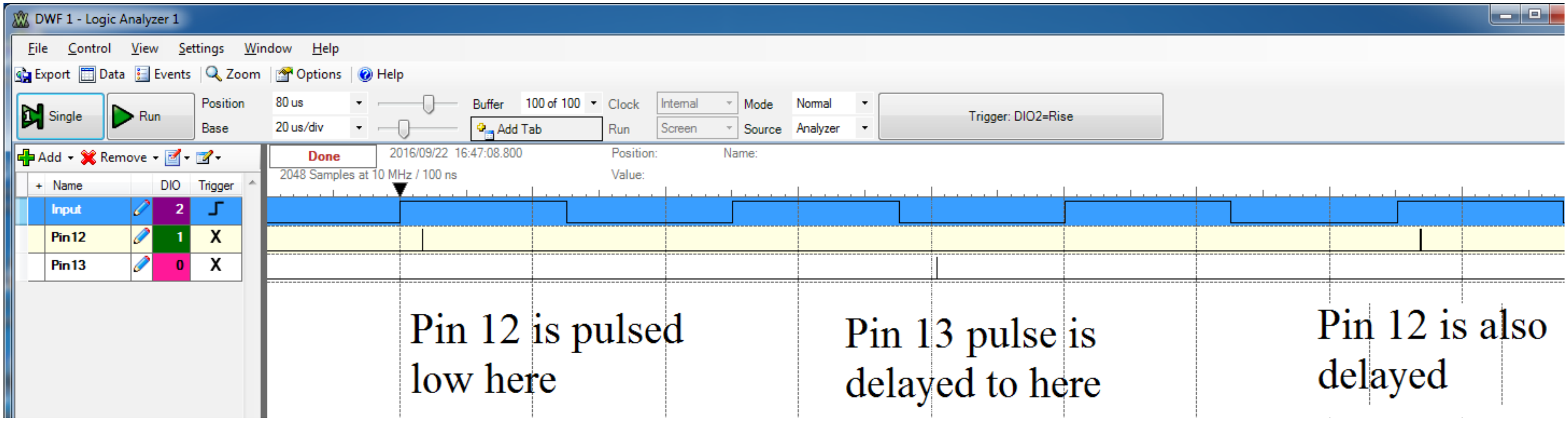
Pin 12 pulsed
on LOW to HIGH

Pin 13 pulsed
on HIGH to LOW

Second State Machine

```
// Create a set of states and actions
// for simple state machine.
enum PStates {
        High, LowWait0, HighWait0,
        Low, LowWait1, HighWait1
};
enum PActions { None, Pulse13, Pulse12 };
// Create state and action for SM
PStates PulseState = High;
PActions PulseAction = None;

// This function will handle the state machine,
// returning an action that is to be done.
PActions PulseNextState(int Input)
{
        // This is the action to be returned,
        PActions ReturnValue = None; // defaults to No action
        // Switch statement for state machine
        switch (PulseState)
        {
          case High: // State indicating input has been high.
                if (Input == LOW) // then once it goes low
                {
                        ReturnValue = Pulse13; // load action into return value.
                        PulseState = LowWait0; // and change state.
                }
                break;
          case LowWait0: // State indicating input has been low.
                // but no response expected
                if (Input == HIGH) // then once it goes high
                {
                        PulseState = HighWait0;     // and change state.
                }
                break;
          case HighWait0: // State indicating input has been high.
                if (Input == LOW) // then once it goes low
                {
                        PulseState = Low; // and change state.
                }
                break;
          case Low: // State indicating input has been low.
                if (Input == HIGH) // then once it goes high
                {
                        ReturnValue = Pulse12; // load action
                        PulseState = HighWait1;     // and change state.
                }
                break;
          case HighWait1: // State indicating input has been high.
                if (Input == LOW) // then once it goes low
                {
                        PulseState = LowWait1; // and change state.
                }
                break;
          case LowWait1: // State indicating input has been low.
                // but no response expected
                if (Input == HIGH) // then once it goes high
                {
                        PulseState = High;     // and change state.
                }
                break;
        } // End of State switch

        return ReturnValue;

} // End of PulseNextState

// put your setup code here, to run once:
void setup() {
        DDRB |= 0x30; // Set 13 and 12 to outputs
        bitClear(DDRB, 1); // DDRB &= 0xFD; // Set 9 to input.
        PORTB &= 0x30; // &= 0xCF; // Set 13 and 12 to LOW.
} // End of setup
```

```
// put your main code here, to run repeatedly:
void loop() {
        PulseAction = PulseNextState(bitRead(PINB, 1));
        switch (PulseAction)
        {
          case Pulse13: // Call to pulse pin 13.
               bitSet(PORTB, 5); // PORTB |= 0x20; // Force high
               bitClear(PORTB, 5); // PORTB &=~0x20; // Force low
               break;
          case Pulse12: // Call to pulse pin 12.
               bitSet(PORTB, 4); // PORTB |= 0x10; // Force high
               bitClear(PORTB, 4); // PORTB &=~0x10; // Force low
               break;
          case None:
               break;

        } // End of Action switch

} // End of loop.
```



Position 80 us

Base 20 us/div

Buffer 100 of 100

Clock Internal

Run Screen

Mode Normal

Source Analyzer

Trigger: DIO2=Rise

2016/09/22 16:47:08.800

2048 Samples at 10 MHz / 100 ns

Position:

Value:

Name:

+ Name	DIO	Trigger
Input	2	⎍
Pin12	1	X
Pin13	0	X

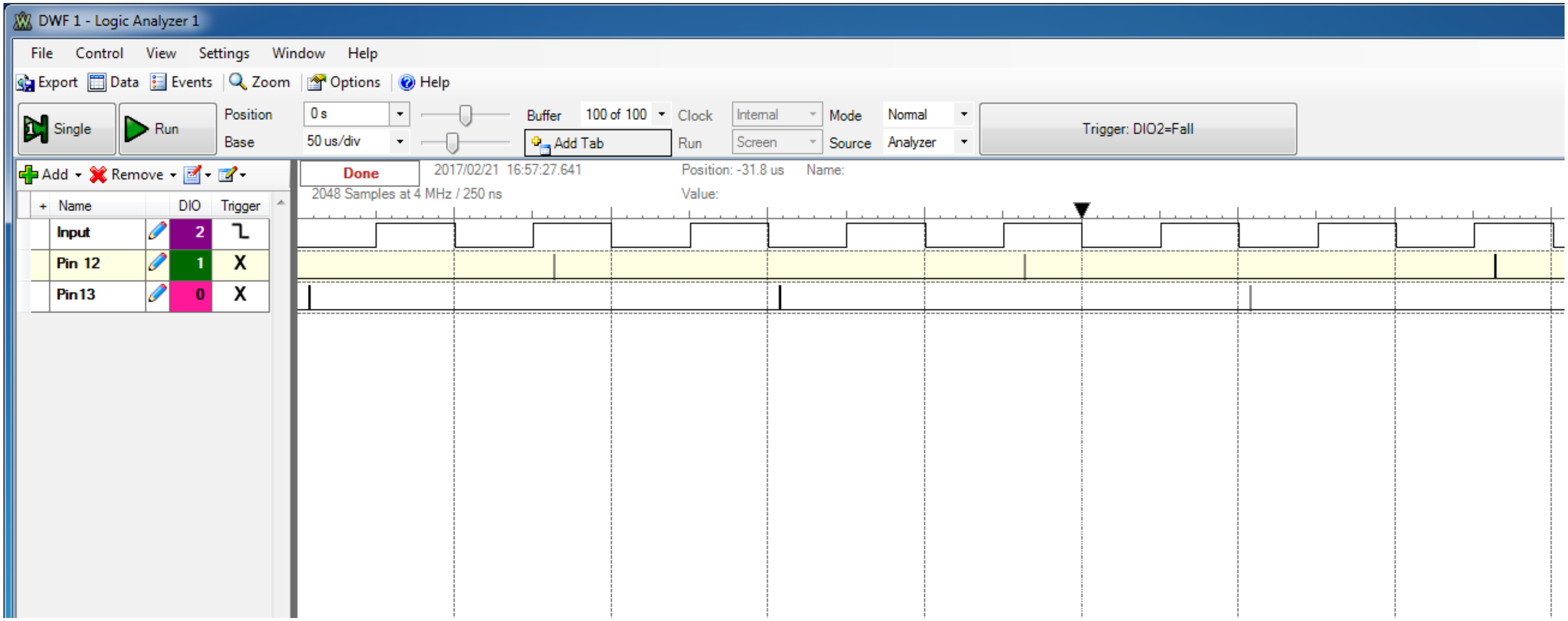
Pin 12 is pulsed low here

Pin 13 pulse is delayed to here

Pin 12 is also delayed

Second State Machine

```
// global variables that will watch for pulses.
int OldInput = HIGH;
int PulseCounter = 2;

void setup()
{
       // put your setup code here, to run once:
       bitSet(DDRB, 5); // Set 13
       bitSet(DDRB, 4); // and 12 to outputs

       bitClear(DDRB, 1); // Set 9 to input.

       PORTB &= 0xCF; // Set 13 and 12 to LOW.
} // end of setup.

// Code that is called continuously.
void loop()
{
       int Input;
       // Read in our input.
       Input = digitalRead(9);

       // Check for a change in
       if (OldInput != Input)
       {
              // Check if count is zero.
              if (PulseCounter != 0)
              {
                     // if not Decrement count
                     PulseCounter--;
              }
              else
              {
                     PulseCounter = 2;  // Reset count
                     // if input is high,
                     if (Input == HIGH)
                     {
                            bitSet(PORTB, 4); // Pulse pin 12;
                            bitClear(PORTB, 4);
                     }
                     else
                     {
                            bitSet(PORTB, 5); // Pulse pin 13;
                            bitClear(PORTB, 5);
                     } // end of if input high test.

              }// end of if checking for count zero.

       } // end of if checking for a change

       // Hold Input value for next pass.
       OldInput = Input;

} // end of loop.
```



Position 0 s

Base 50 us/div

Buffer 100 of 100

Clock Internal

Run Screen

Mode Normal

Source Analyzer

Trigger: DIO2=Fall

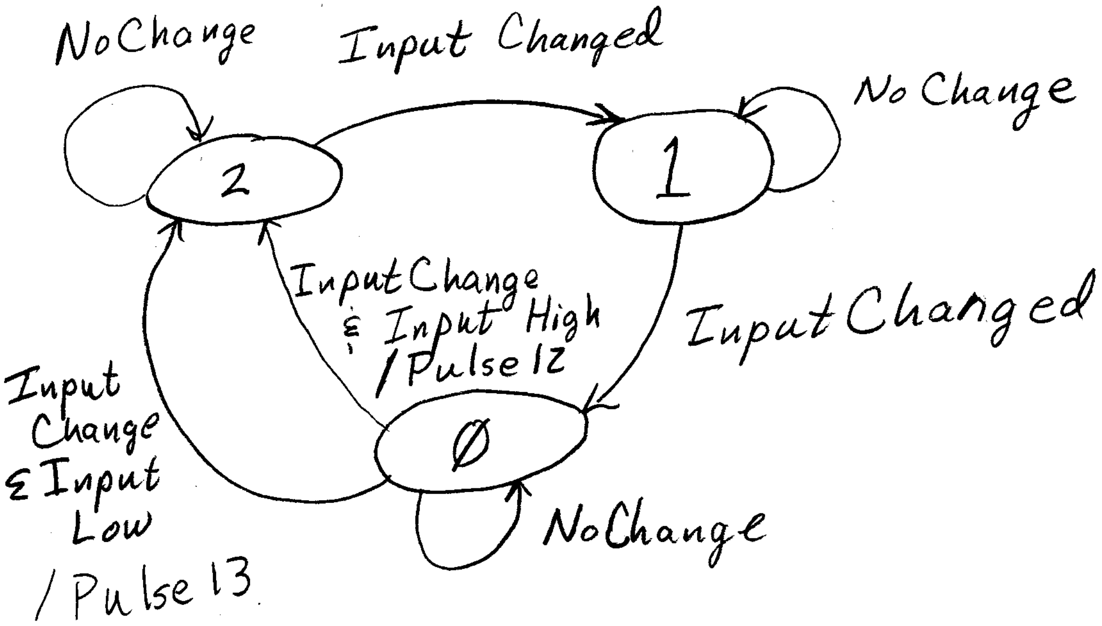
Done 2017/02/21 16:57:27.641

2048 Samples at 4 MHz / 250 ns

Position: -31.8 us Name:

Value:

+	Name	DIO	Trigger
	Input	2	⅂
	Pin 12	1	X
	Pin13	0	X


NoChange
Input Changed
No Change
2
1
Input Change
& Input High
/Pulse 12
Input Changed
Input
Change
& Input
Low
/Pulse 13
Ø
NoChange