Configurable Logic Cell (CLC)

TIPS 'N TRICKS INTRODUCTION

Microchip continues to provide innovative products that are smaller, faster, easier to use and more reliable. Flash-based PIC® microcontrollers (MCUs) are used in a wide range of everyday products from smoke detectors to industrial, automotive and medical products.

The PIC16(L)F150X and PIC10(L)F32X families of devices with on-chip configurable logic cells merge all the advantages of the PIC® MCU architecture and the flexibility of Flash program memory with the functionality of a configurable digital logic cell. Together, they form a low-cost building block with resource savings and external component reduction.

The flexibility of Flash and an excellent development tool suite, including a low-cost In-Circuit Debugger, In-Circuit Serial Programming™ (ICSP™) and CLC Configuration Tool GUI, make these devices ideal for just about any embedded control application.

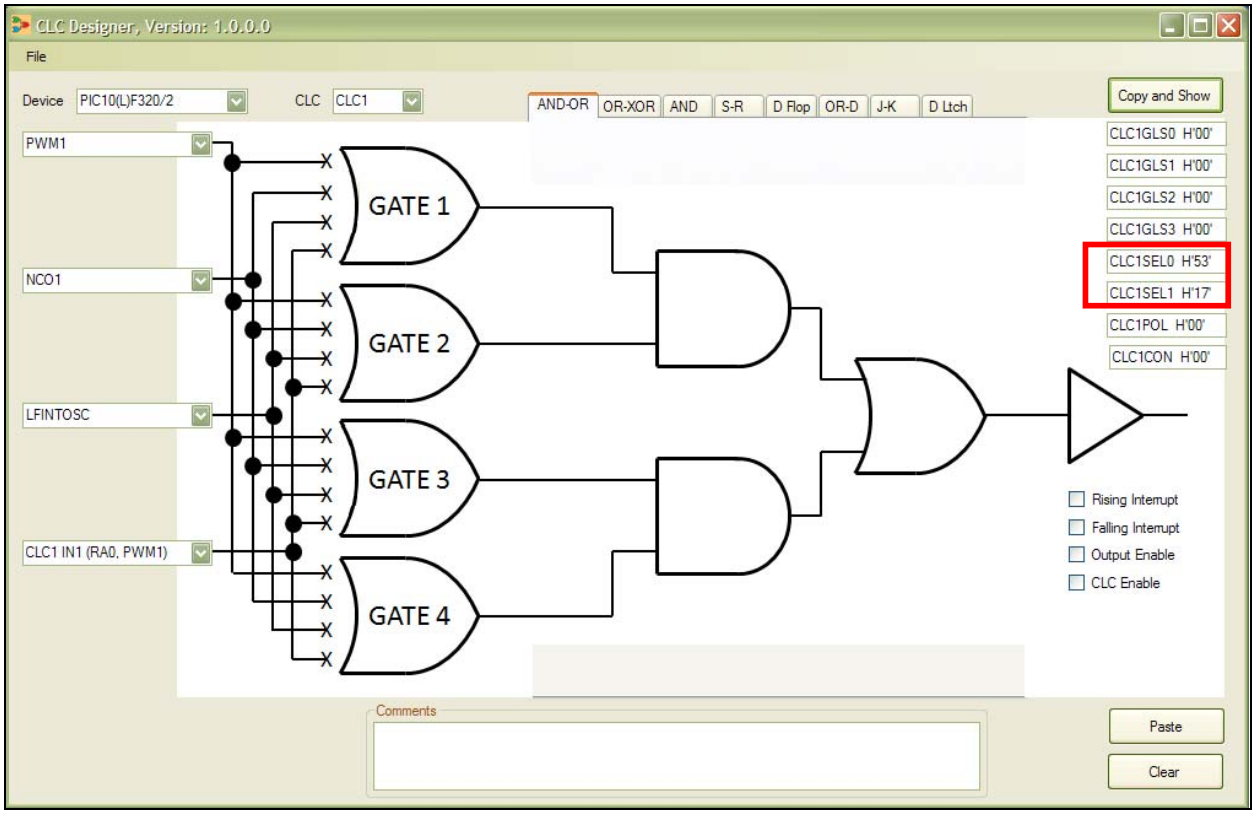
The following series of Tips 'n Tricks can be applied to a variety of applications to help make the most of digital logic functions using a PIC MCU with on-chip configurable logic.

CLC OVERVIEW

Input Selection

For all CLC modules, there are eight signals available as inputs to the configurable logic cell, and these eight input signals may vary from device to device. Nevertheless, only four can be selected at any one time. This is done via four 8-input multiplexers, used to pass the input signals on to the data gating stage of the CLC. Input signals are selected with the CLCxSEL0 and CLCxSEL1 registers, as shown in Figure 1.

FIGURE 1: CLC CONFIGURATION



Data Gating

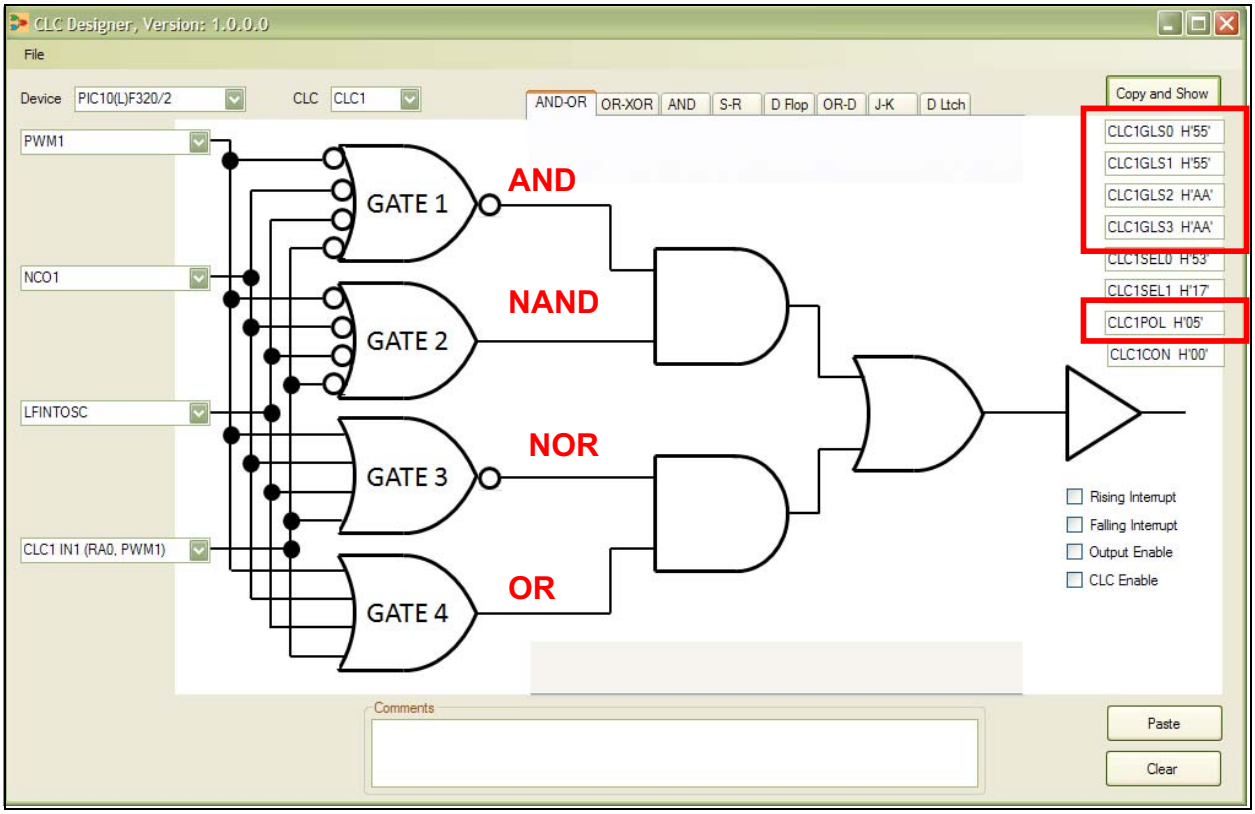
The outputs from the input multiplexers are directed to the data gating stage of the CLC. The data gates can be configured to direct each input signal as inverted or non-inverted data signals. These signals are then ANDed together in each gate. Finally, each gates output can be inverted before going on to the logic function stage of the CLC.

The basic logic that can be obtained in each gate is summarized in Table 1 and Figure 2.

TABLE 1: DATA GATING LOGIC

	CLCxGLS(0-3) Registers	LCxGyPOL bits	Gate Logic
Inverted	55h	1	AND
	55h	0	NAND
Non-Inverted	AAh	1	NOR
	AAh	0	OR
Not Connected	00h	0	Logic 0
	00h	1	Logic 1

FIGURE 2: CLC DATA GATING LOGIC



LOGIC FUNCTION SELECTION

The outputs from the four data gates are now inputs into the logic function selection stage of the CLC. Here, the data gate outputs can be gated down to one output signal from a selection of eight logic functions. These eight logic functions are shown in Figure 3 through Figure 10.

FIGURE 3: AND-OR

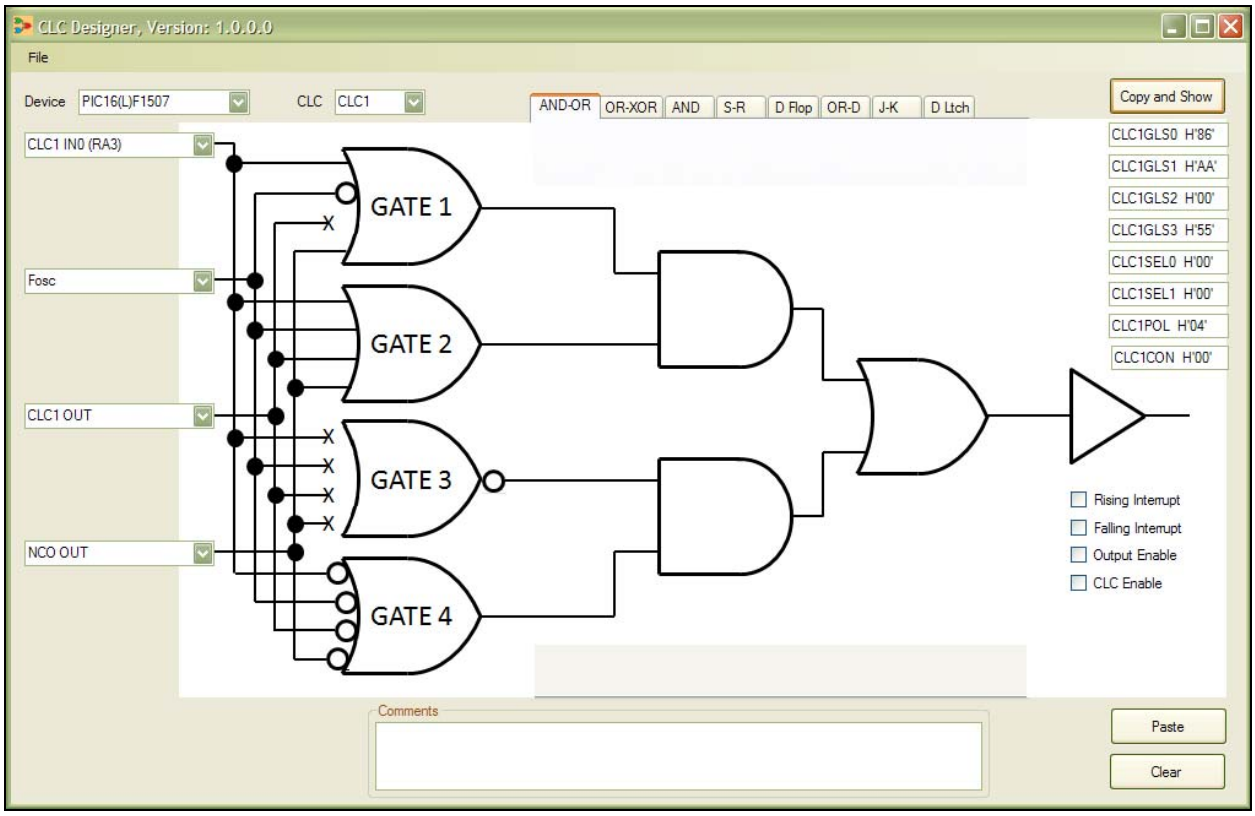


FIGURE 4: OR-XOR

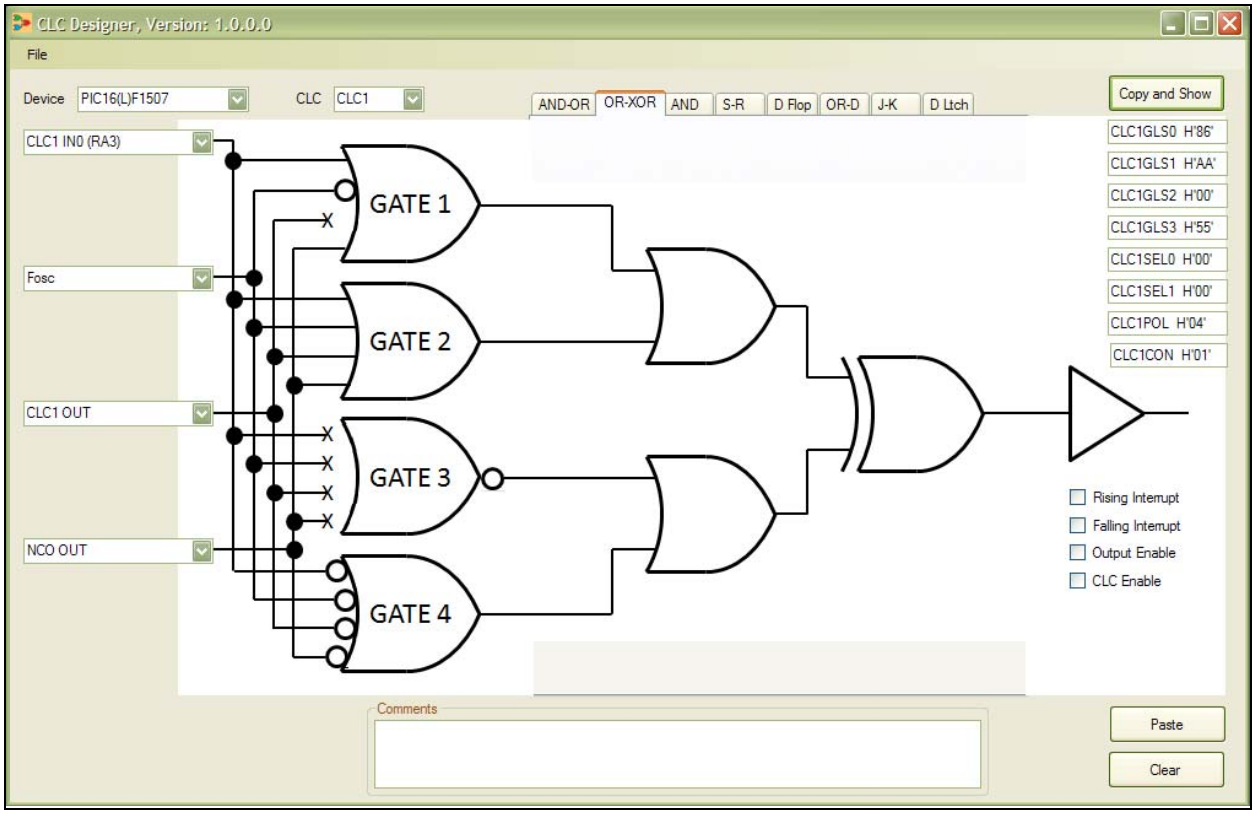


FIGURE 5: 4 – INPUT AND

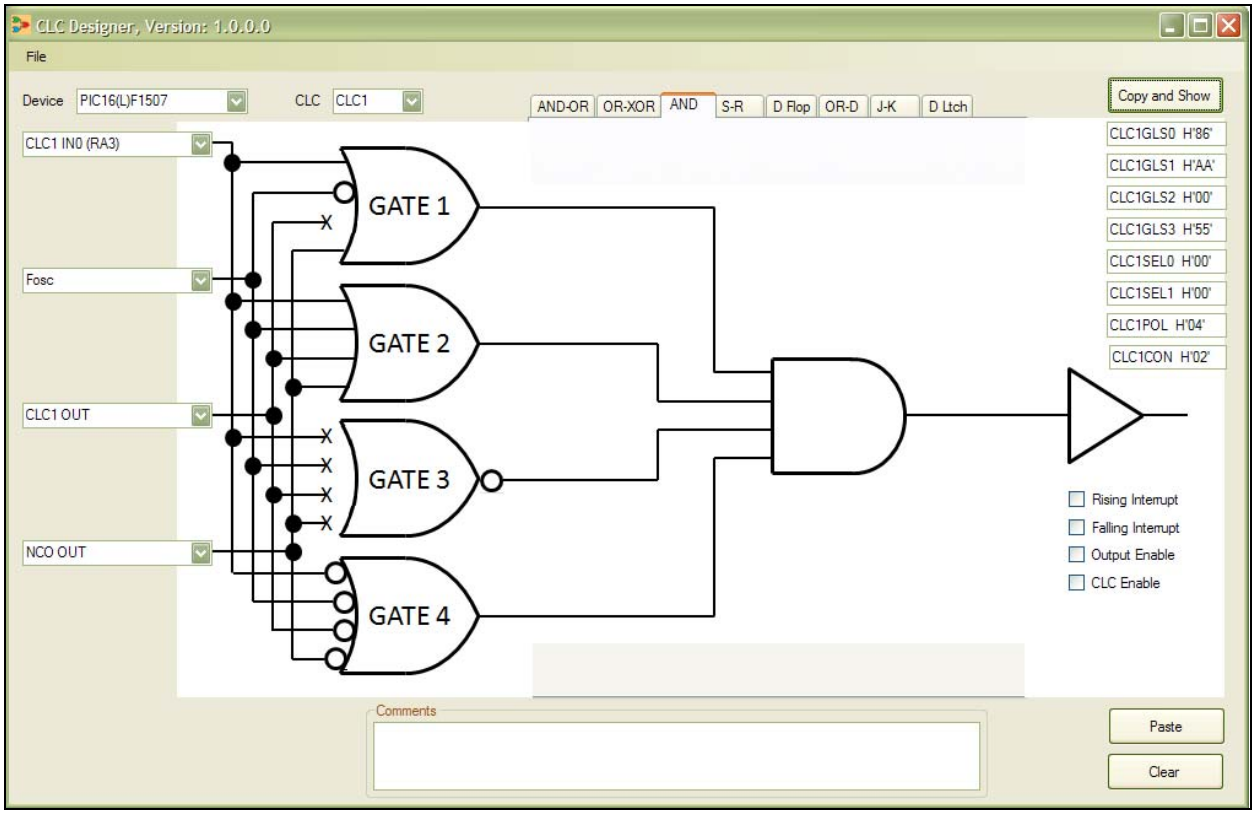


FIGURE 6: SR LATCH

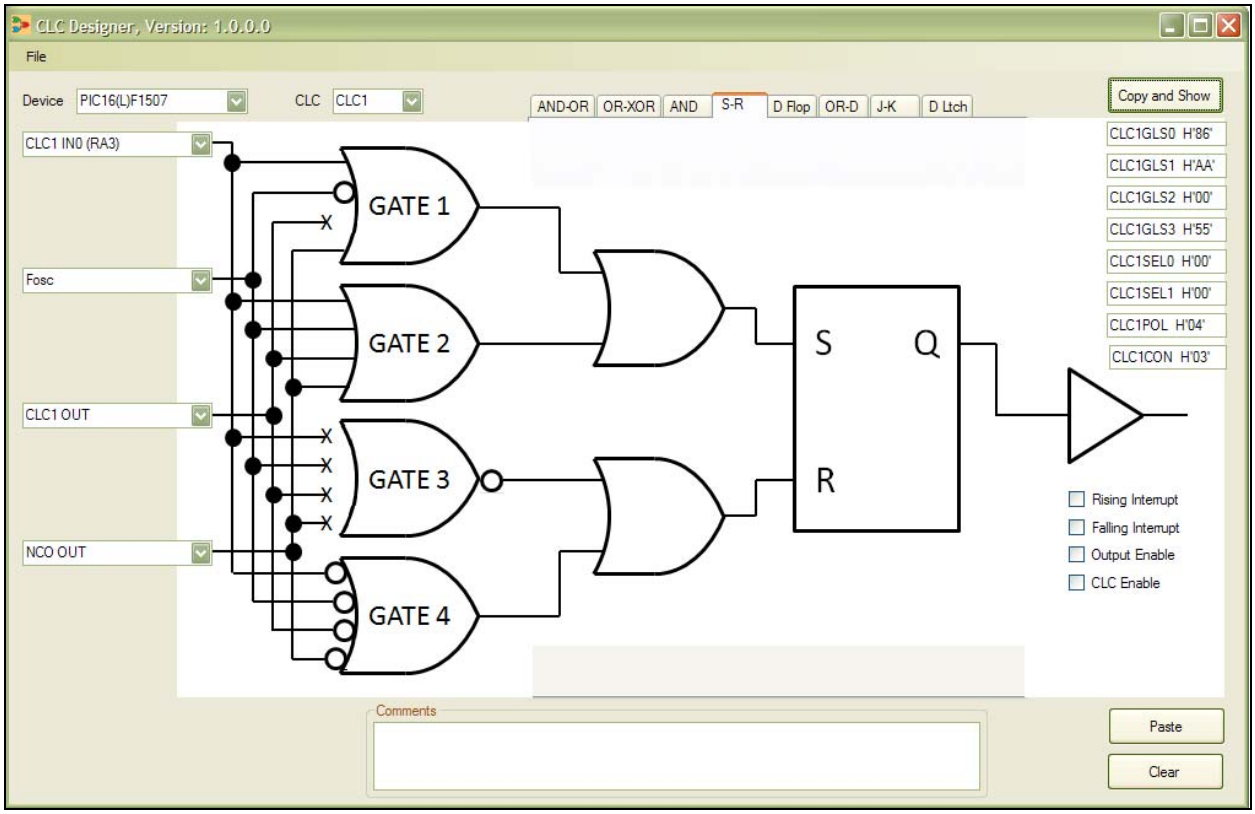


FIGURE 7: 1 – INPUT D FLIP-FLOP WITH SET AND RESET

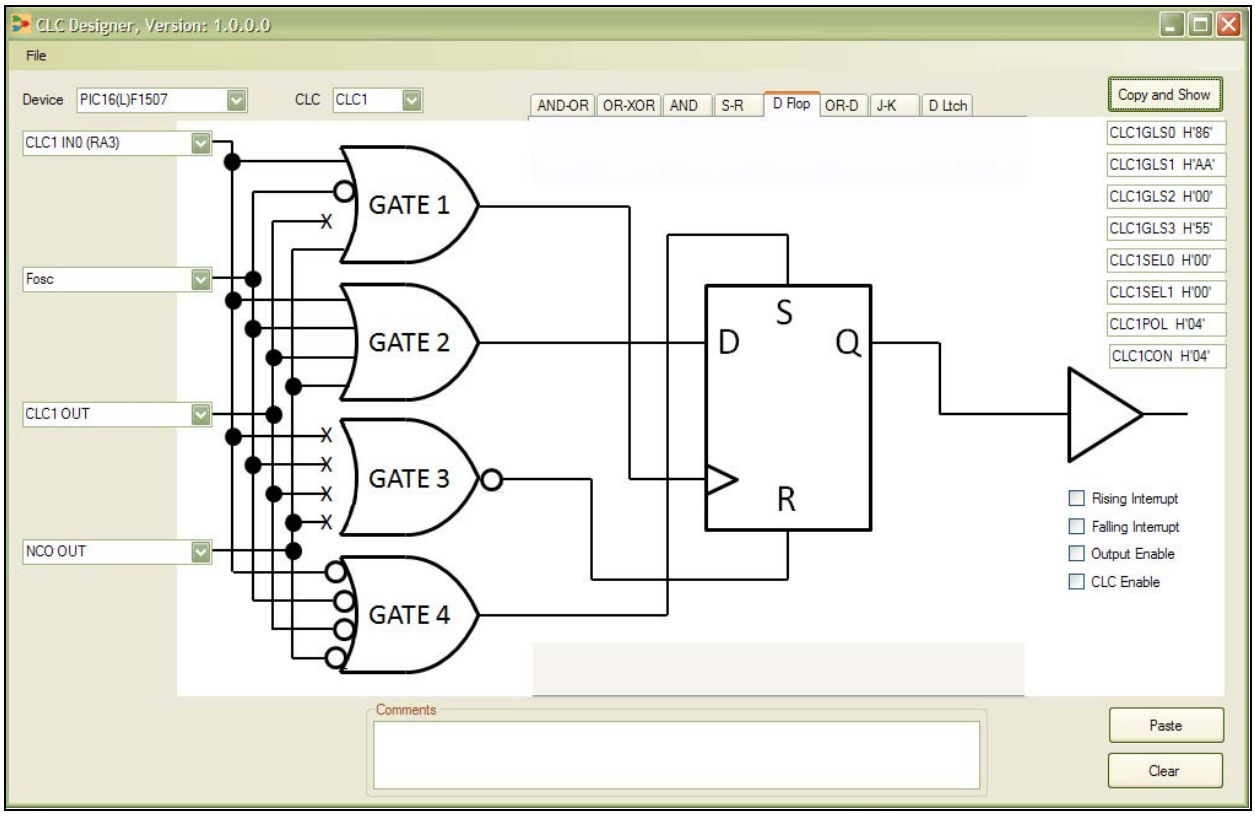


FIGURE 8: 2 – INPUT D FLIP-FLOP WITH RESET

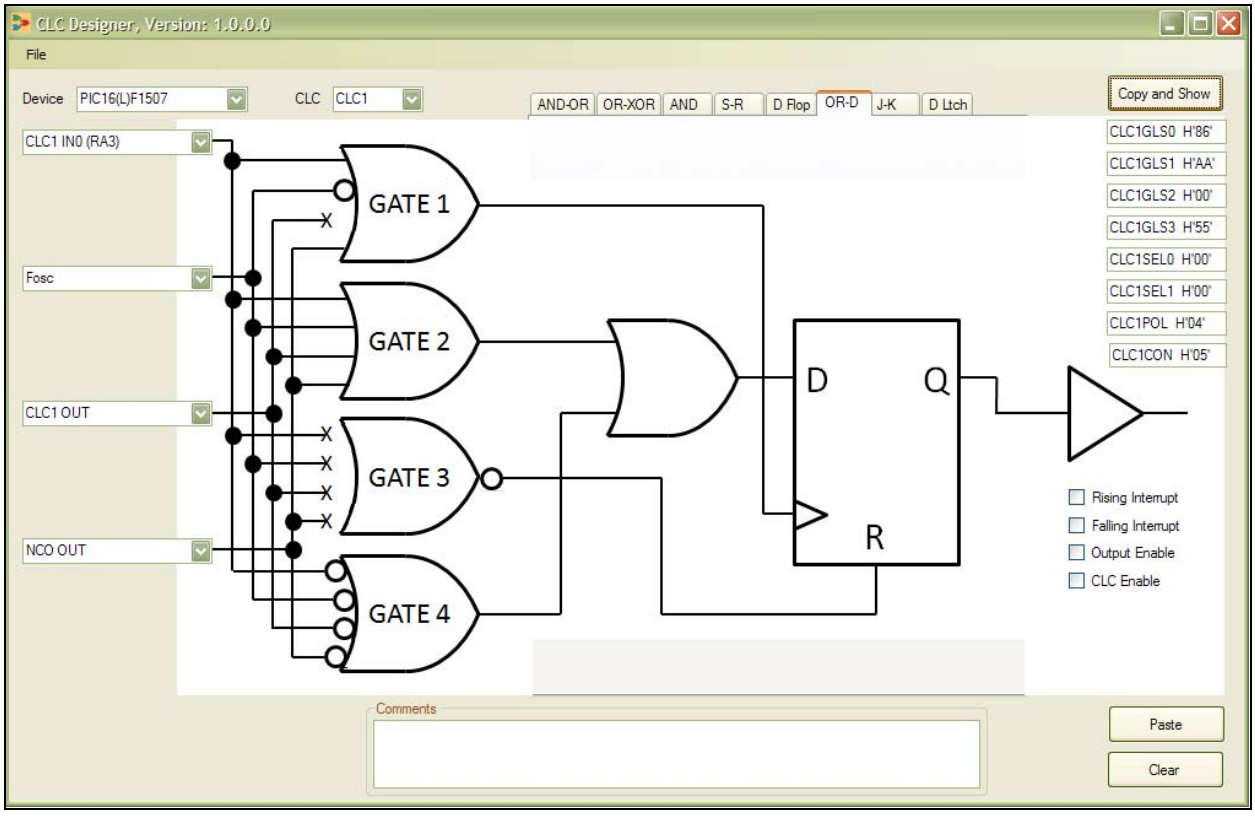


FIGURE 9: J – K FLIP-FLOP WITH RESET

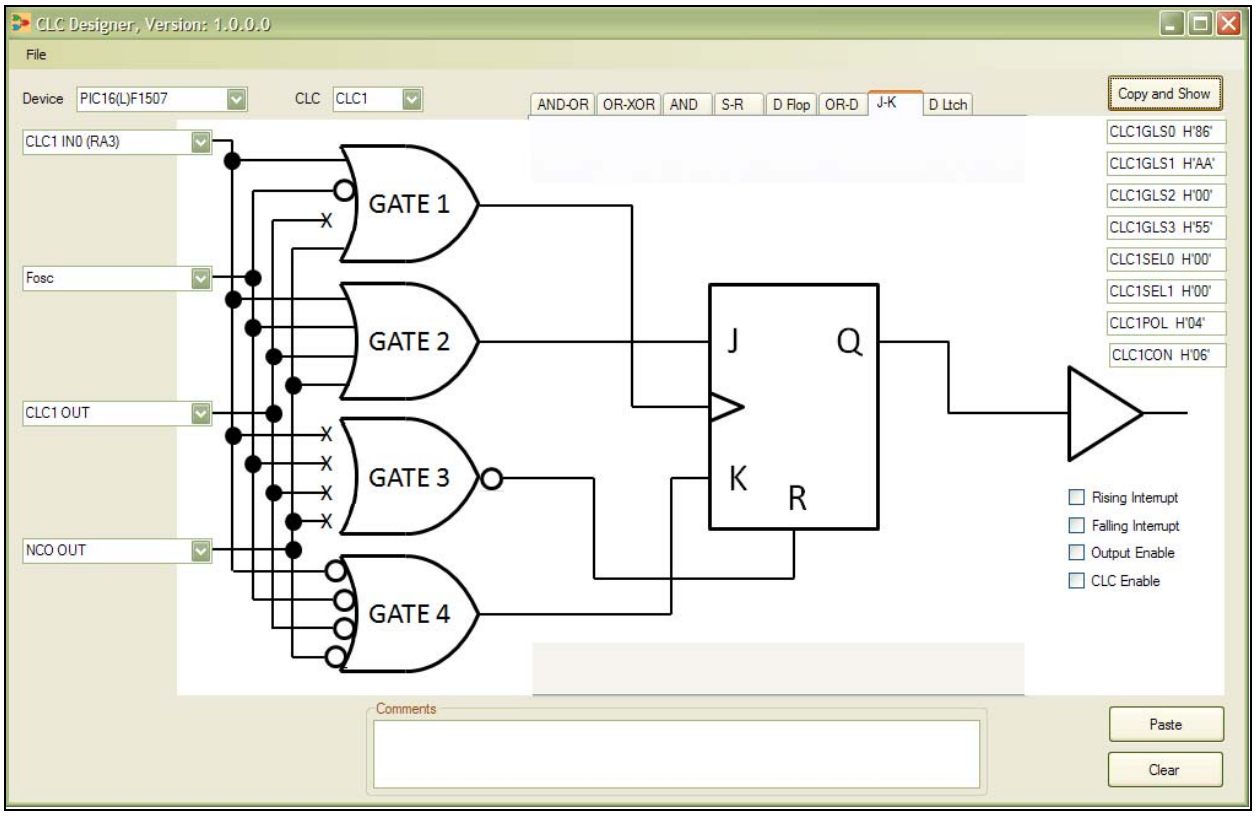
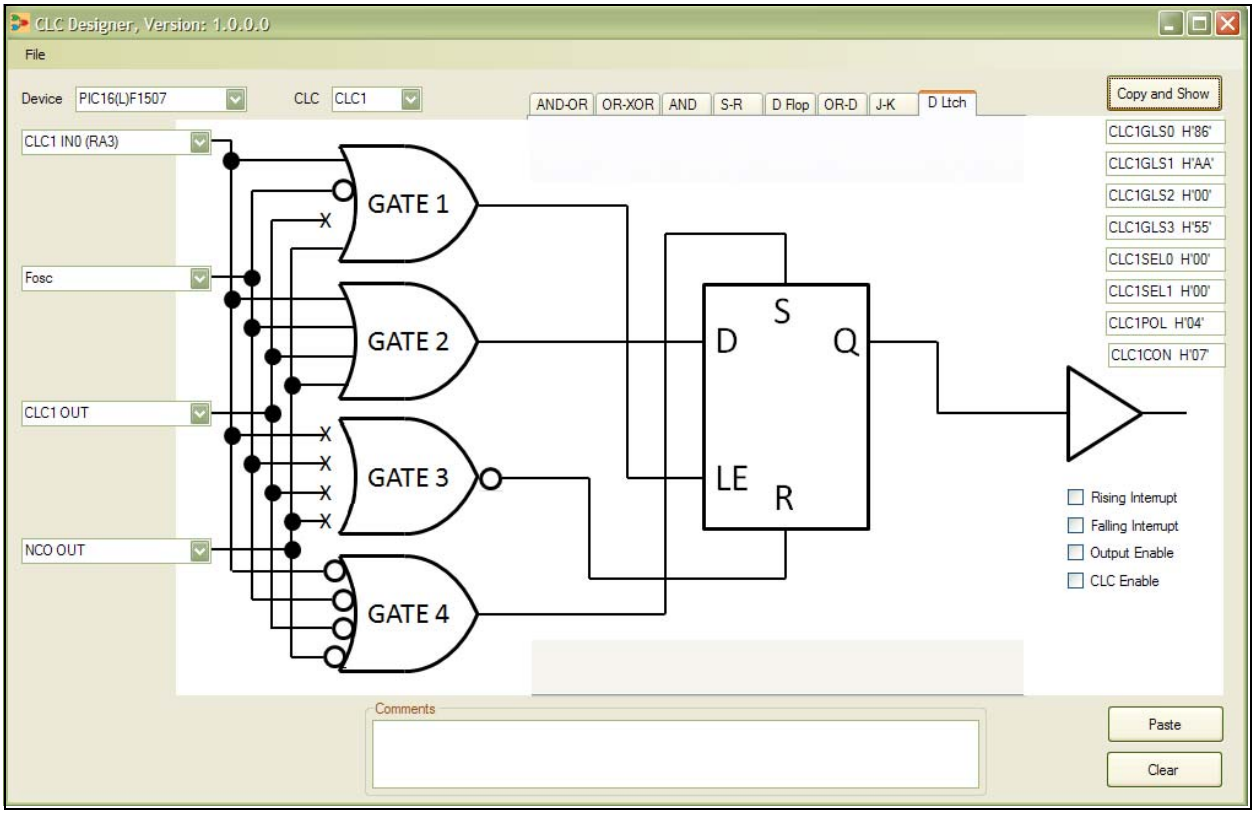


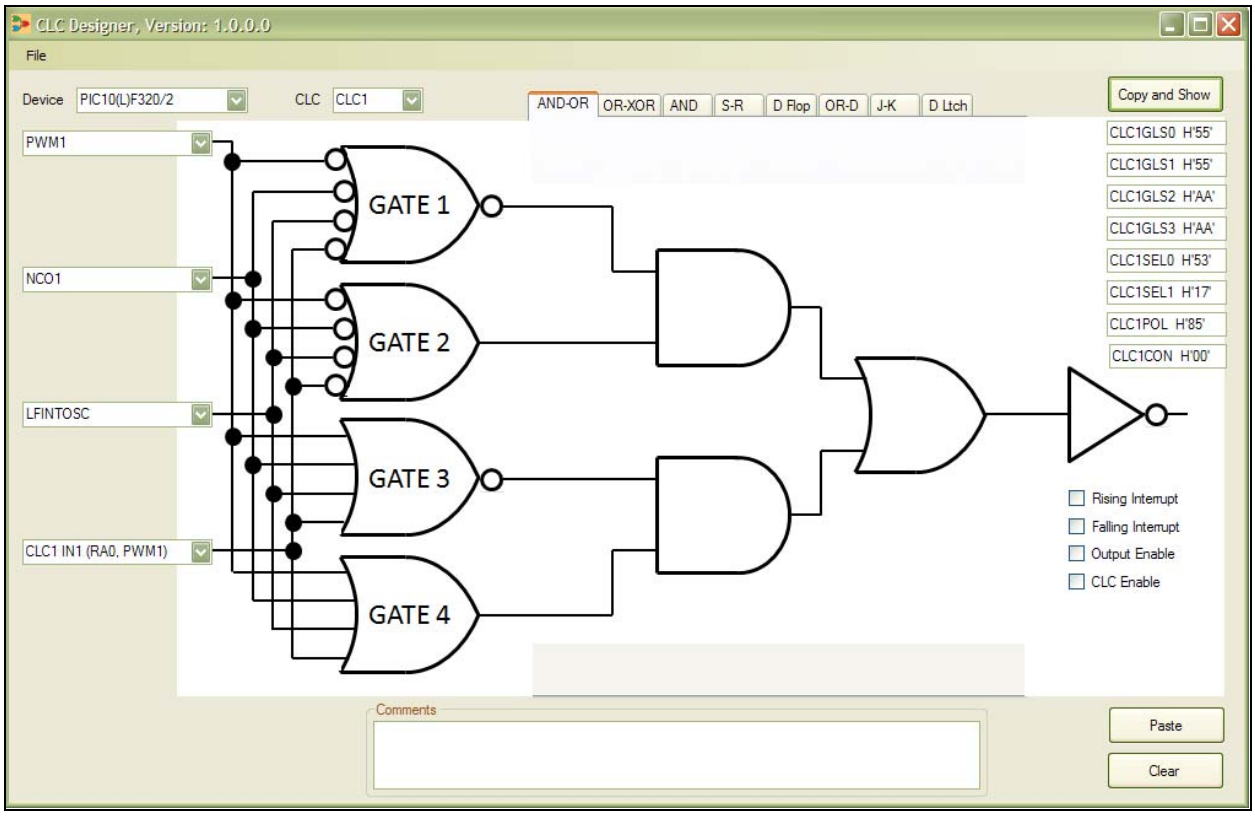
FIGURE 10: 1 – INPUT TRANSPARENT LATCH WITH SET AND RESET



OUTPUT CONTROL

The last stage of the CLC is the output control stage. Here the output signal from the logic function selection stage can be inverted, or not, and sent to the device output pin or sent internally to other peripherals. Also, an interrupt can be generated upon a change in the CLC output signal. This interrupt can be set to occur on the rising or falling edge of the CLC output signal.

FIGURE 11: 1 – INPUT TRANSPARENT LATCH WITH SET AND RESET



TIP 1: REROUTING AN OUTPUT PIN

How often have you needed to move a signal on one pin of a PIC MCU to another? Often, this will have to be done by extra source code that eats up device resources or by physically adding a jumper wire that does not look very good. This tip presents a simple method of rerouting one device pin to another pin on the same device internally, using the CLC module without using up precious resources.

Although there are some limits placed on which pins can be routed to which locations, the input pin must be one of the CLC inputs and the destination pin must be one of the CLC output pins.

TABLE 2: EXAMPLE ALLOCATION TABLE

I/O	20-Pin PDIP/SOIC/SSOP	20-Pin QFN	A/D	Reference	CWG	NCO	CLC	Timers	PWM	Interrupt	Pull-up	Basic
RA0	19	16	AN0	—	—	—	—	—	—	IOC	Y	ICSPDAT
RA1	18	15	AN1	VREF+	—	—	—	—	—	IOC	Y	ICSPCLK
RA2	17	14	AN2	—	$\overline{\text{CWG1FLT}}$	—	CLC1(1)	T0CKI	PWM3	INT/ IOC	Y	—
RA3	4	1	—	—	—	—	CLC1IN0	—	—	IOC	Y	$\overline{\text{MCLR}}$ VPP
RA4	3	20	AN3	—	—	—	—	$\overline{\text{T1G}}$	—	IOC	Y	CLKOUT
RA5	2	19	—	—	—	NCO1CLK	—	T1CKI	—	IOC	Y	CLKIN
RB4	13	10	AN10	—	—	—	—	—	—	IOC	Y	—
RB5	12	9	AN11	—	—	—	—	—	—	IOC	Y	—
RB6	11	8	—	—	—	—	—	—	—	IOC	Y	—
RB7	10	7	—	—	—	—	—	—	—	IOC	Y	—
RC0	16	13	AN4	—	—	—	CLC2	—	—	—	—	—
RC1	15	12	AN5	—	—	NCO1(1)	—	—	PWM4	—	—	—
RC2	14	11	AN6	—	—	—	—	—	—	—	—	—
RC3	7	4	AN7	—	—	—	CLC2IN0	—	PWM2	—	—	—
RC4	6	3	—	—	CWG1B	—	CLC2IN1	—	—	—	—	—
RC5	5	2	—	—	CWG1A	—	CLC1(2)	—	PWM1	—	—	—
RC6	8	5	AN8	—	—	NCO1(2)	—	—	—	—	—	—
RC7	9	6	AN9	—	—	—	CLC1IN1	—	—	—	—	—
VDD	1	18	—	—	—	—	—	—	—	—	—	VDD
VSS	20	17	—	—	—	—	—	—	—	—	—	VSS

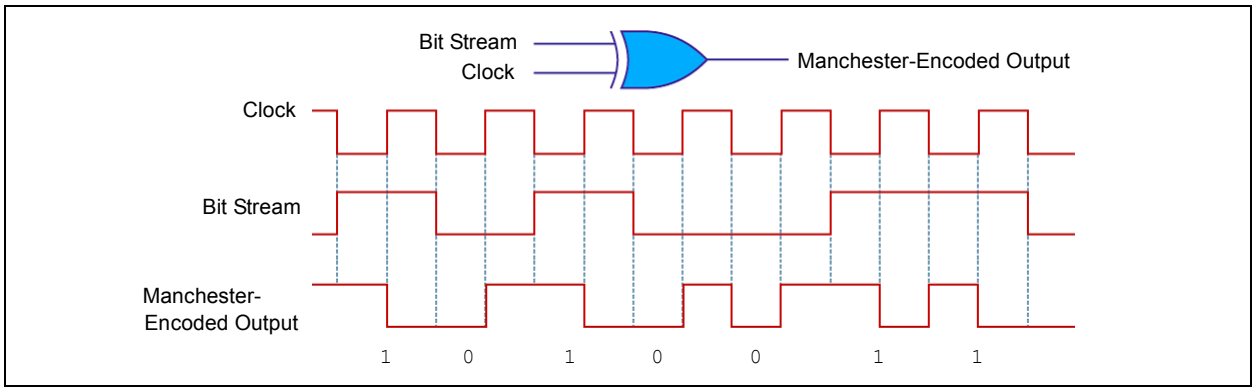
Note 1: Default location for peripheral pin function. Alternate location can be selected using the APFCON register.
2: Alternate location for peripheral pin function selected by the APFCON register.

This device has two CLC modules, see Table 2. CLC1 and CLC2. CLC1 has inputs on pins RA3 and RC7, either one of these pins can be moved to RA2 or RC5 (RC5 requires the use of the APFCON register). Likewise, when using CLC2, it has inputs on pins RC3 and RC4, either one of these pins can be moved to RC0 only.

TIP 2: MANCHESTER ENCODER

Manchester encoding is a versatile line encoding method, which is widely used. When the EUSART is used to transmit data, various mechanisms such as interrupts and buffers are available to free up resources on the CPU. To date, however, it was required to either perform a bit-bang transmission or use external hardware to take this output signal and encode it in Manchester format.

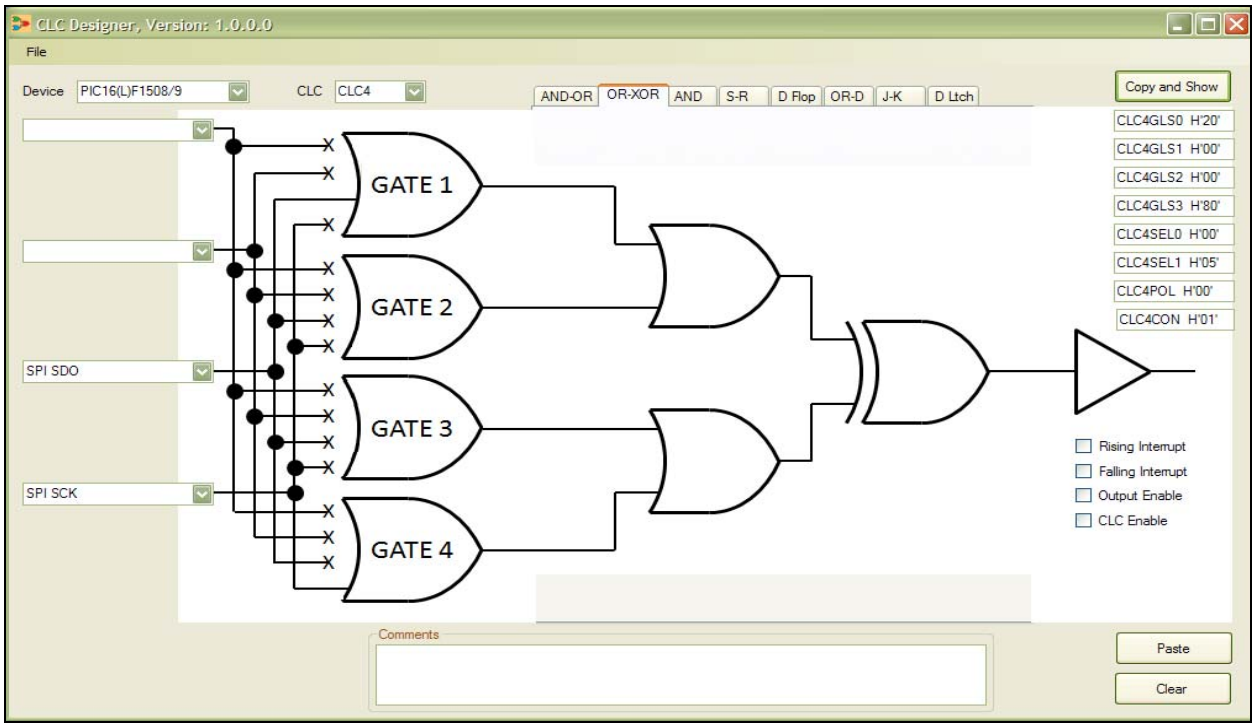
FIGURE 12: MANCHESTER-ENCODED SIGNAL



This tip presents a method to produce a Manchester-encoded output signal by using the SPI port in Synchronous mode with the CLC. By combining the SPI clock with the SPI data using the CLC, a Manchester-encoded signal can be created in hardware with no overhead and no external components required to do the modulation.

Configure the CLC as shown in Figure 13. The output will be a Manchester-encoded version of the data sent via SPI in Master mode.

FIGURE 13: CLC CONFIGURED FOR MANCHESTER ENCODING

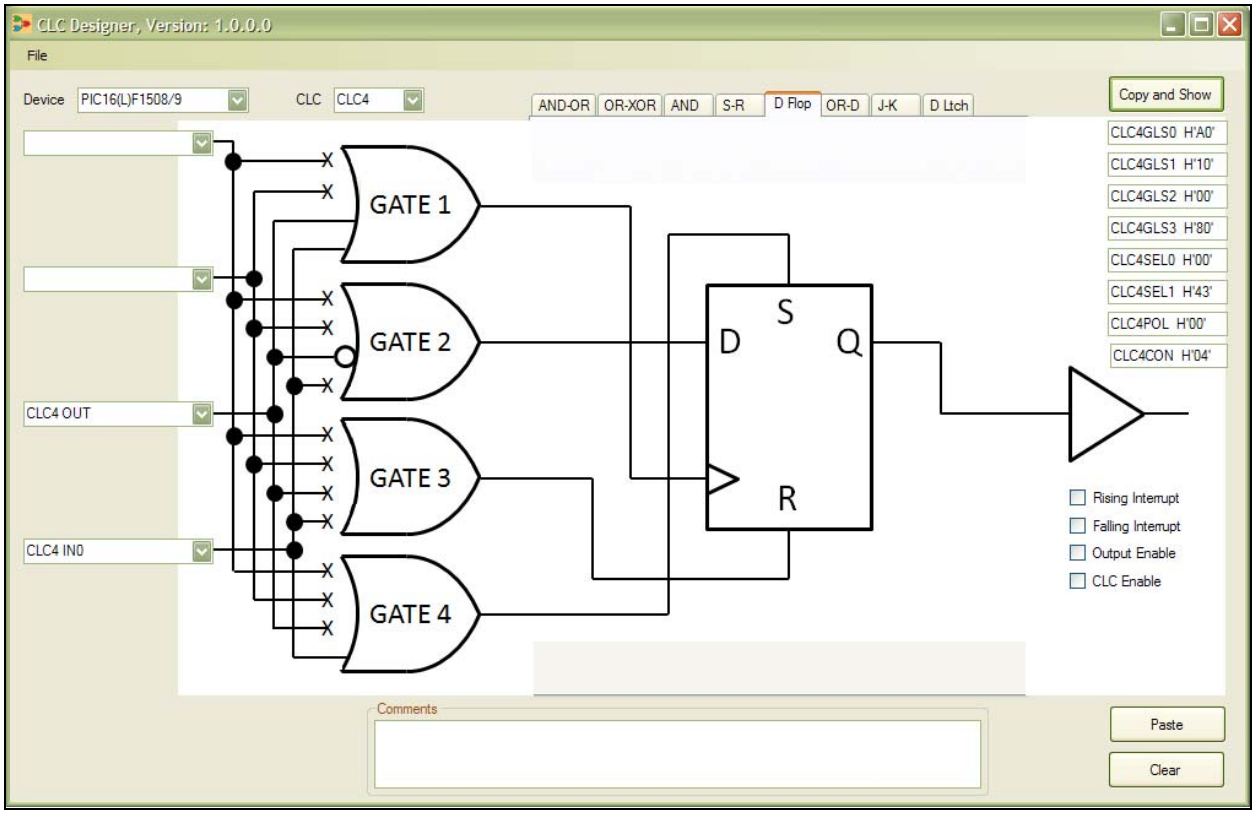


TIP 3: FREQUENCY DIVIDER

Frequency dividers are commonly used building blocks for more complex applications.

By negating the CLC4OUT signal when feeding it into D, we are effectively tapping out $\overline{Q}$. The flip-flop clocks this input through to the output on the positive edge of the next external clock, causing the output to toggle once for every positive edge coming in from the external signal. This results in the input clock being divided by two at the output. Using the CLC as a D flip-flop, the user can create a simple frequency divider by connecting it up as follows. See Figure 14.

FIGURE 14: CLC CONFIGURED FOR FREQUENCY DIVIDER



TIP 4: CONDITIONAL WAKE FROM SLEEP

In applications where power use is critical, it is common to put the microcontroller to Sleep in order to save power, and wake it up only when a specific event has occurred which requires attention.

If the condition we are looking for requires a number of signals to represent a specific state, it often results in the CPU waking from Sleep due to a pin change, only to check the condition and realize that the other inputs, which constitute the specific condition, have not occurred, resulting in a waste of power.

This tip describes how to wake the microcontroller from Sleep when a combination of things are true.

Since the CLC keeps running even when the device has been placed in Sleep mode, and the device can be woken from Sleep by an interrupt created by a CLC output changing, it is possible to conditionally wake the device from Sleep. The CLC can be configured to perform a number of logical operations such as OR, XOR or AND operations on input signals, and even combine this with stateful behavior by incorporating flip-flops, waking the device from Sleep only when a very specific combination occurs.

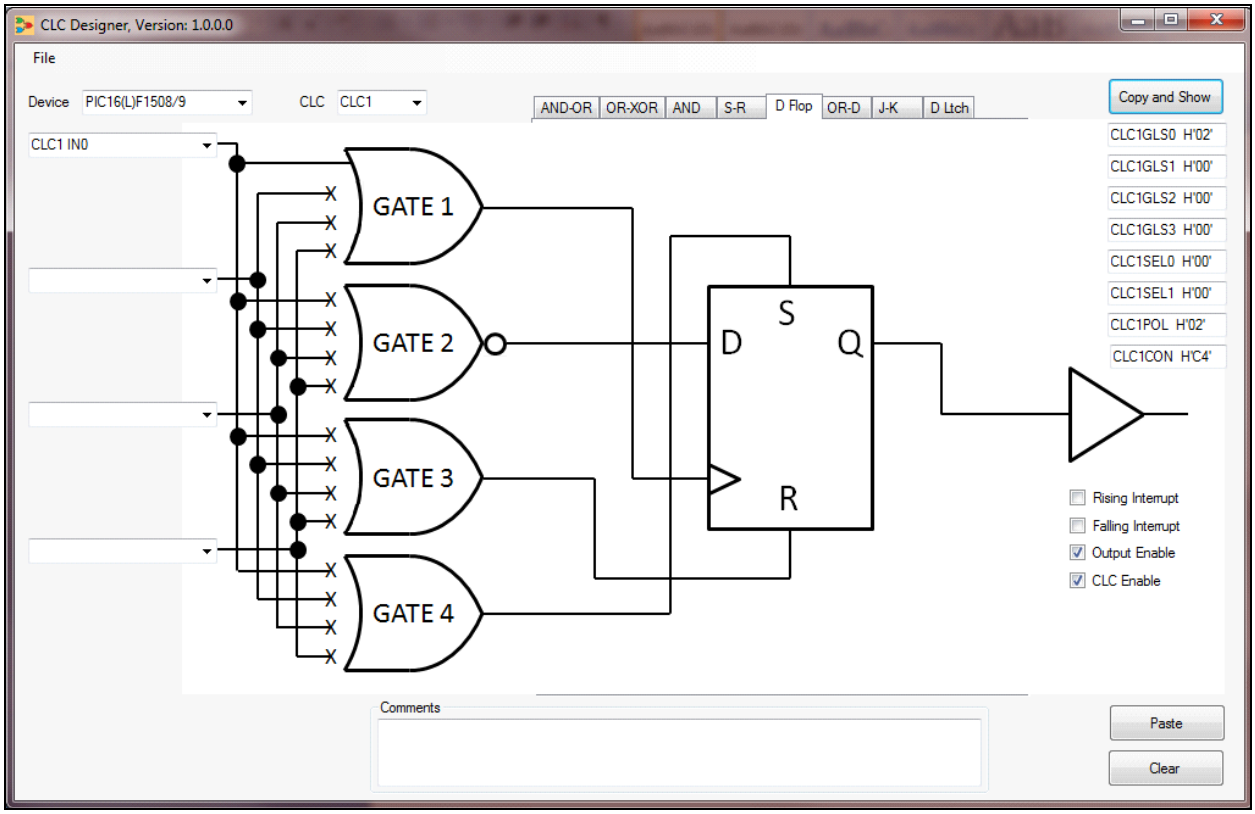
TIP 5: FAST PULSE DETECTOR/ PULSE EXTENDER

When using a microprocessor to do pulse counting or simply react to a condition where an input pin is presented with a very short one-shot pulse, it is often a problem that these small pulses are missed, resulting in incorrect behavior.

While it is possible to solve this problem by using an interrupt-on-change mechanism, many applications have to operate in deterministic time (real time) and are thus prevented from using interrupts. In this case, inputs need to be polled at a specific time to determine the value, making it impractical to count short pulses.

This tip describes a way to detect a clock edge on an external pin and hold it, even if the input changes back to the original state very quickly. By Configuring the CLC to clock the pulse edge into a D flip-flop, as shown in Figure 15, it is possible to save the pulse for an indefinite amount of time, allowing the microprocessor to read and react to the impulse at its own leisure.

FIGURE 15: CLC CONFIGURED TO CLOCK A D FLIP-FLOP



This will solve the problem in all cases where multiple pulses are not expected in quick succession. This same technique also allows for the debouncing of a contact that needs to be read, ensuring that multiple events are not triggered by a single contact change.

Variation:

By adding an input to the reset line of the D flip-flop and feeding this back from the output via a RC filter, it is possible to simply extend the pulse instead of continuing the signal indefinitely.

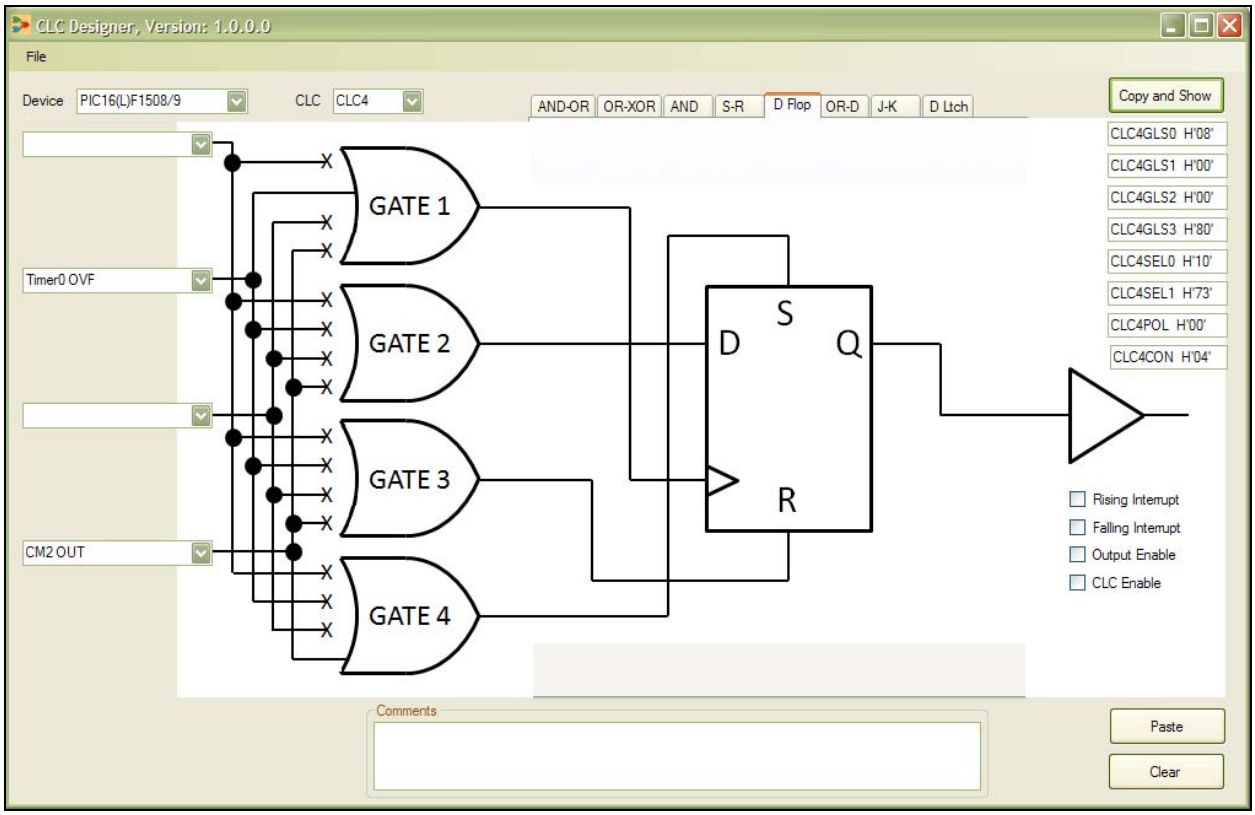
TIP 6: SIGNAL THRESHOLD AND HOLD CIRCUIT

Interfacing a digital device, such as a CPU, with an analog device, such as a photodiode, can be challenging. As the output signal will be offset by the ambient light, which may vary widely between different conditions such as being indoors, or outdoors in direct sunlight. These conditions can cause the entire upper (logic 1), or the entire lower (logic 0) part of the signal to fall within the undefined range on the device (between V_{IN0max} and V_{IN1min}). On many devices, these values can be significantly far apart, as digital electronics are designed to operate at discreet values of ‘1’ and ‘0’, and not in between.

In order to overcome this problem, it is necessary to change the threshold where the decision is made whether the signal represents a ‘0’ or a ‘1’, and eliminate as much as possible of the undefined region in between the two. This can be accomplished by using an on-chip comparator to sample the signal, by feeding the non-inverting input signal to the comparator from an internal Digital-to-Analog Converter (DAC) peripheral.

This tip presents a simple method of sampling the input signal with a precise threshold, overcoming the problem of a signal floating in the undefined region of a normal input pin. The comparator is set up to sample and hold the input signal precisely at the bias point using the internal DAC. The CLC is configured as a D flip-flop to sample and hold the value as follows.

FIGURE 16: CLC CONFIGURED AS A D FLIP-FLOP TO SAMPLE AND HOLD VALUES



For example, when decoding a quadrature-encoded input signal from a optical rotary encoder, a timer can be set up to sample both inputs in this fashion and adjust for the ambient offset by adjusting the bias voltage from the DAC.

TIP 7: QUADRATURE DECODER

Many input devices, such as rotary encoders, provide a quadrature-encoded output signal, which needs to be decoded to determine if the device has been turned and in which direction it has been turned. See Figure 18.

A common problem with circuits that decode this quadrature-encoded signal (see Figure 17) occurs when the input is left between a ‘0’ and a ‘1’, and one of the two lines is toggled repeatedly, causing the device to mistakenly detect the dial is still being turned, while it is in fact, stationary.

FIGURE 17: QUADRATURE-ENCODED SIGNAL

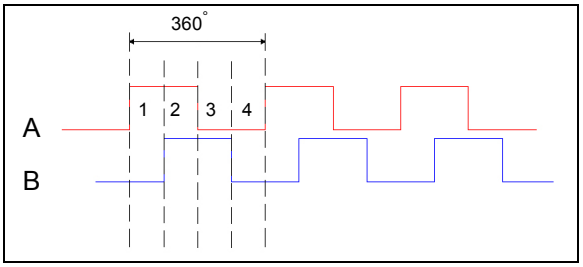
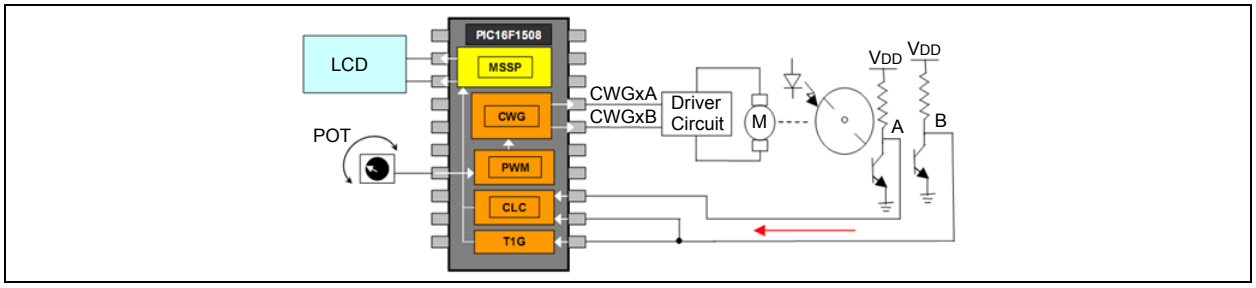


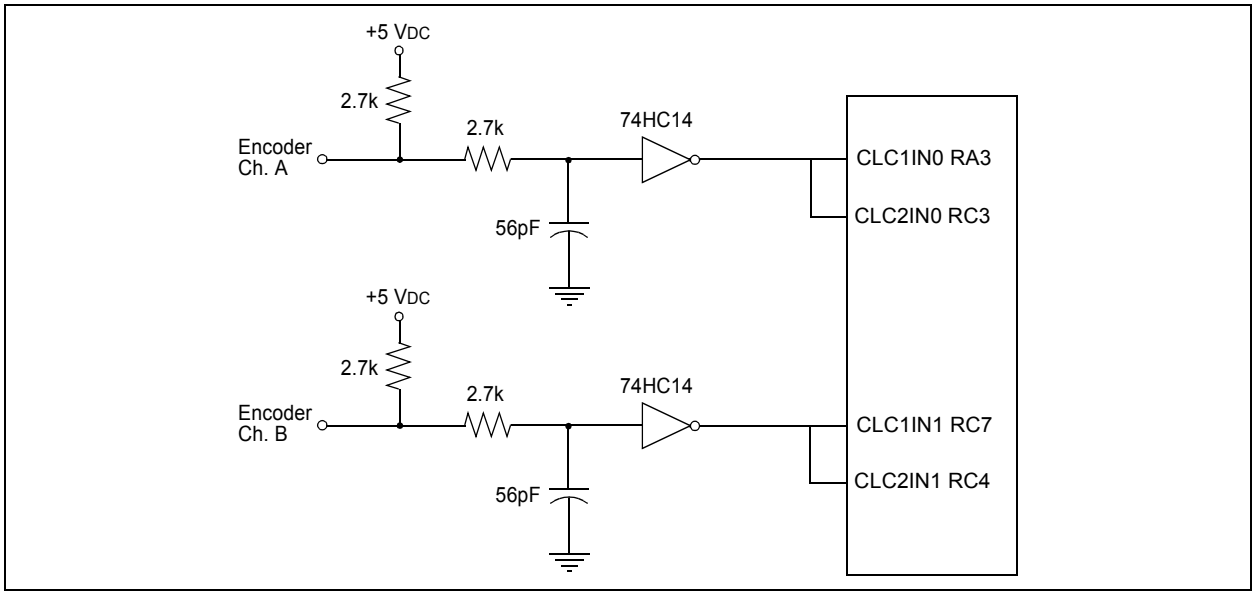
FIGURE 18: TYPICAL QUADRATURE DECODER CIRCUIT



This tip describes how to use the CLC to decode a quadrature-encoded input signal such as a rotary encoder. As seen above, the line connected to the flip-flop clock toggles repeatedly due to noise if it is left between a ‘1’ and a ‘0’ level, and will cause the counter to keep on counting (or “run”) without the turning of the wheel.

The circuit below (Figures 19, 20, and 21) uses two D-type flip-flops with a clear input to generate two separate pulse trains for clockwise and anti-clockwise rotation. By clearing the output from the line, which is not used as the clock, we ensure that the circuit will never “run” in one direction, if the dial is not being turned.

FIGURE 19: SCHEMATIC OF ROTARY ENCODERS CONNECTION TO THE CLC INPUTS



Using the CLC, the D-type flip-flops needed are available on-chip with no external components required. (Note that the CLR input is active-low, so in the CLC this input needs to be configured as inverted between D and CLR.)

FIGURE 20: CLC1 CONFIGURATION FOR ROTARY ENCODER SIGNALS

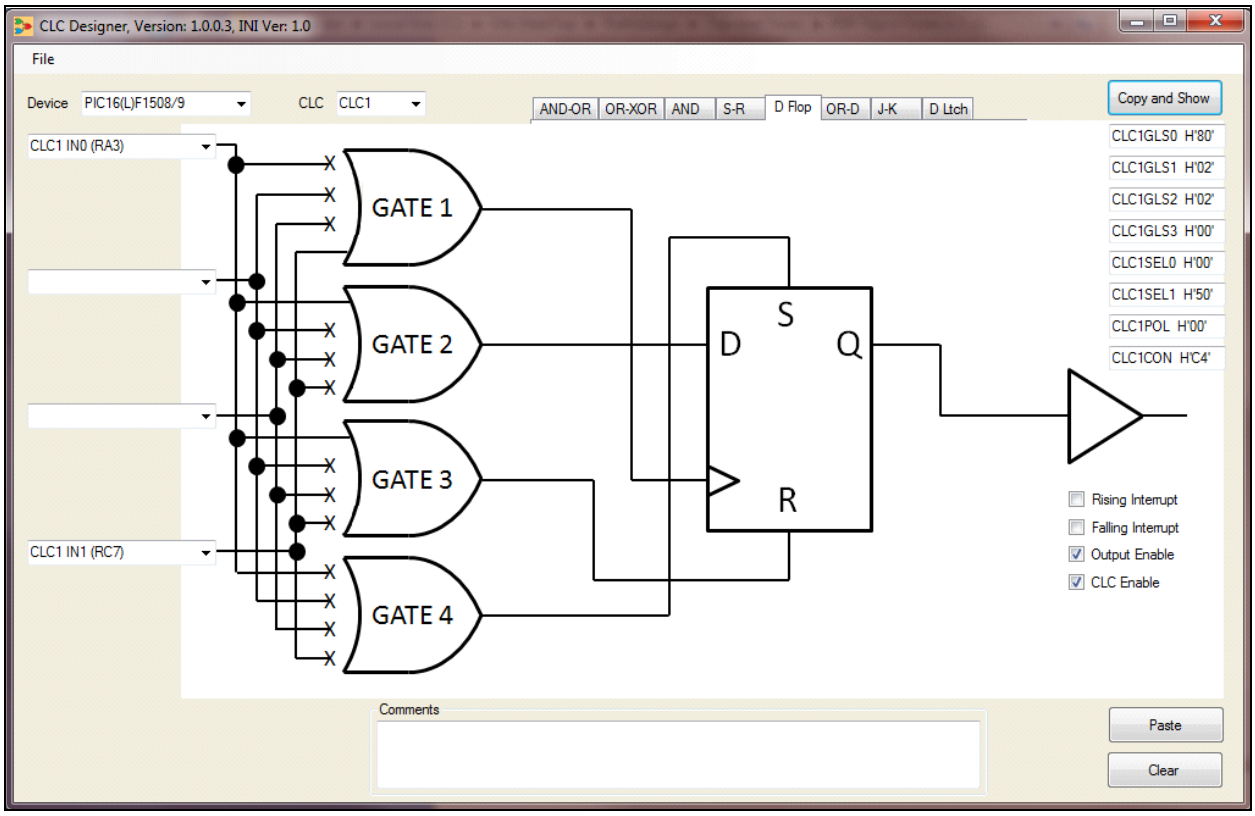
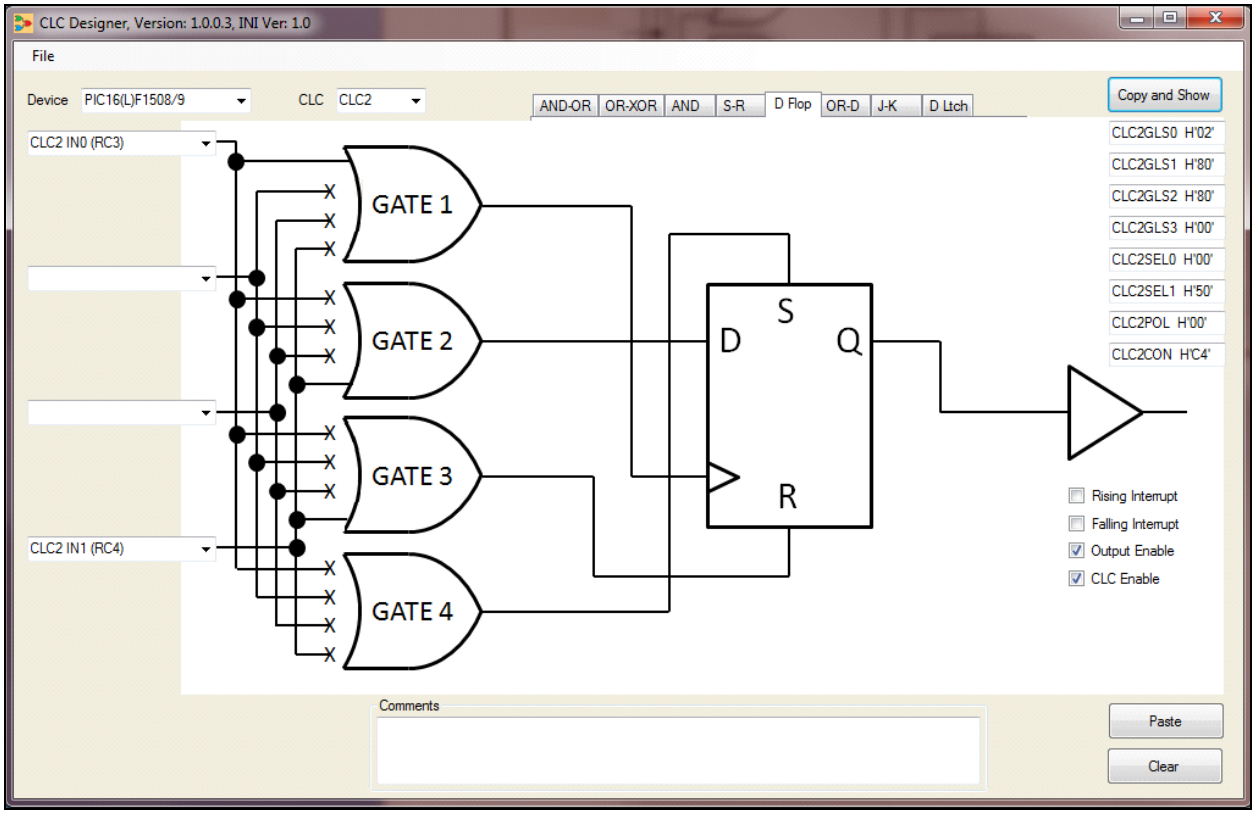


FIGURE 21: CLC2 CONFIGURATION FOR ROTARY ENCODER SIGNALS



TIP 8: PWM STEERING

Pulse-Width Modulation (PWM) applications can be challenging, especially if an application needs one PWM signal in up to four different locations, or up to four different PWM signals in up to four different locations.

This tip describes how to use the CLC to steer one or up to four different PWM signals to up to four different pins on a device.

The first example will show how to set up all four CLC’s to output four different PWM signals. The second example will show how to set up all four CLC’s to output one PWM signal.

EXAMPLE 1

First, you need a device that has four CLC peripherals, like the PIC16F1508. Second, set up the CLC2 with the output of PWM1 as an input, CLC3 with PWM2 as the input, CLC4 with PWM4 as the input, and CLC1 with PWM3 as the input. Then, AND-OR the PWM signal to the specific output pin for each CLC, as shown in Figures 22, 23, 24 and 25.

FIGURE 22: CLC1 CONFIGURATION FOR EXAMPLE 1

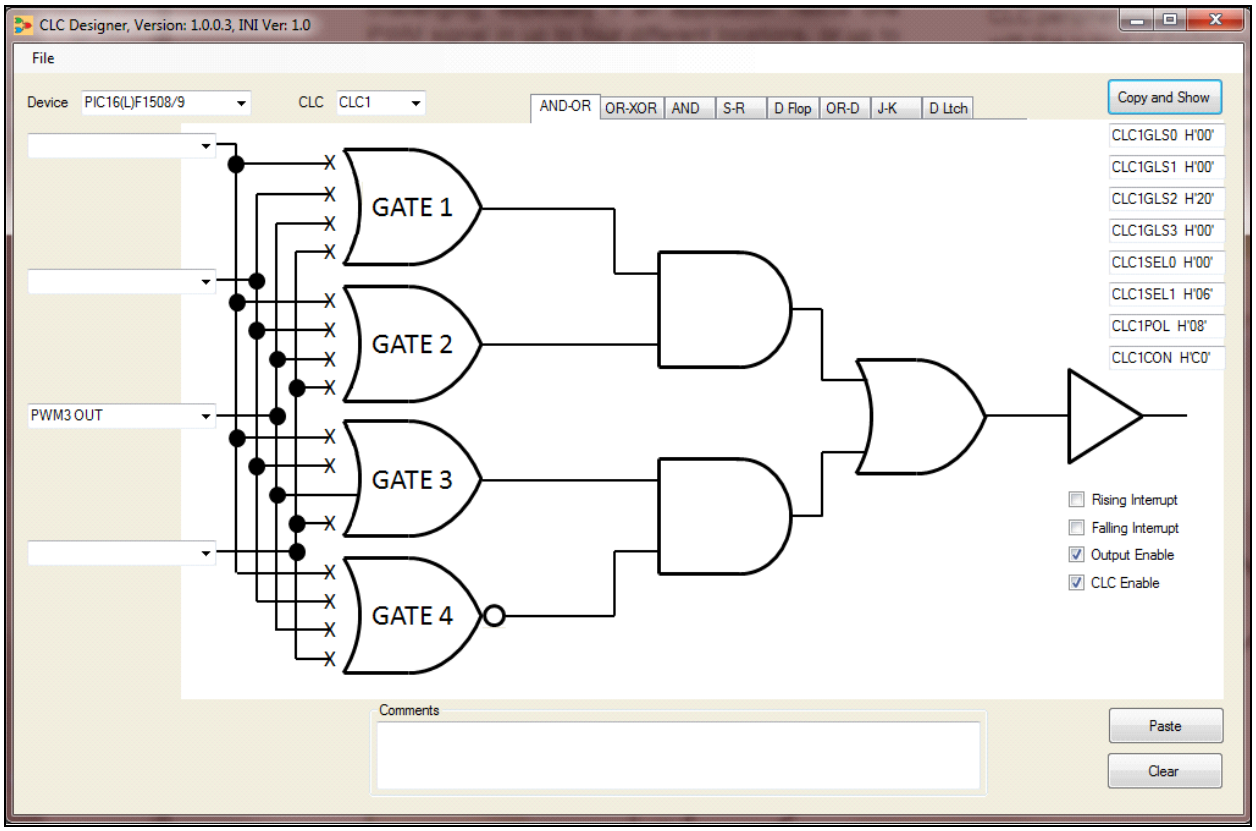


FIGURE 23: CLC2 CONFIGURATION FOR EXAMPLE 1

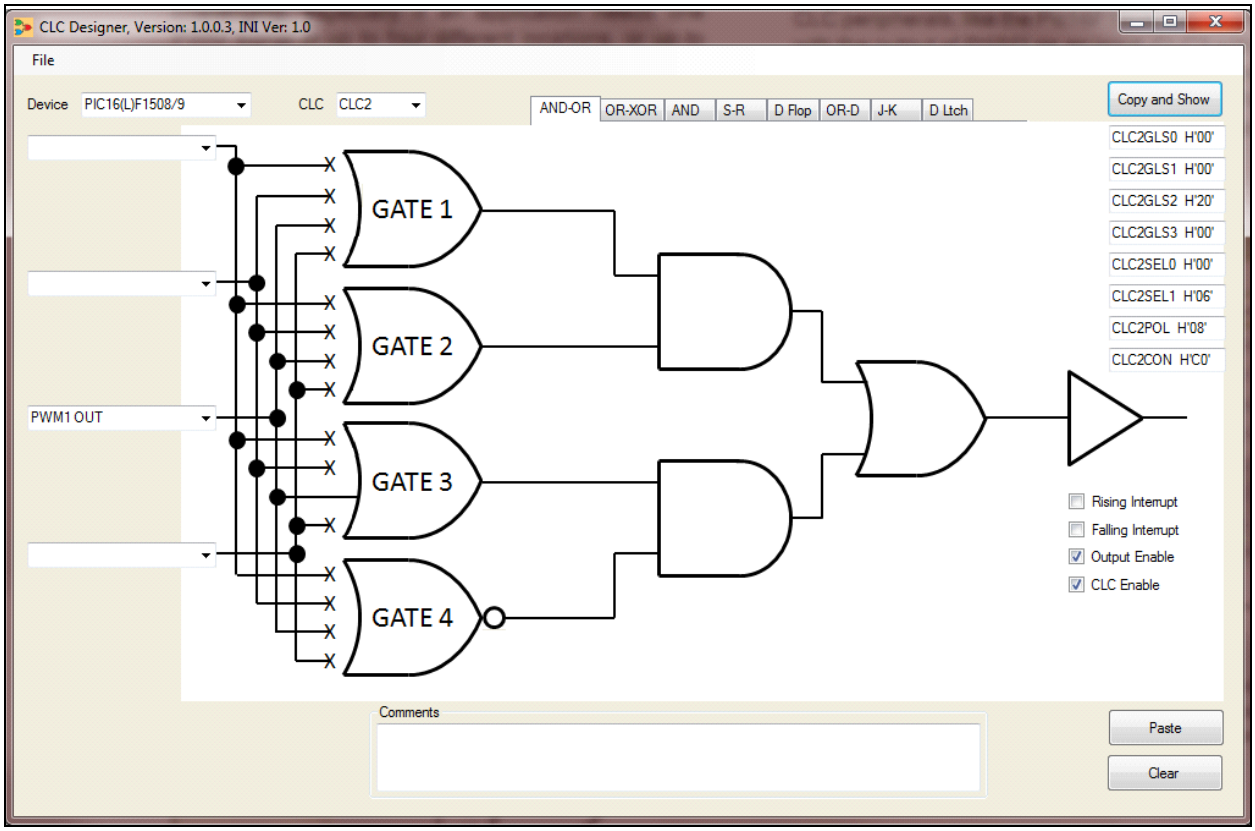


FIGURE 24: CLC3 CONFIGURATION FOR EXAMPLE 1

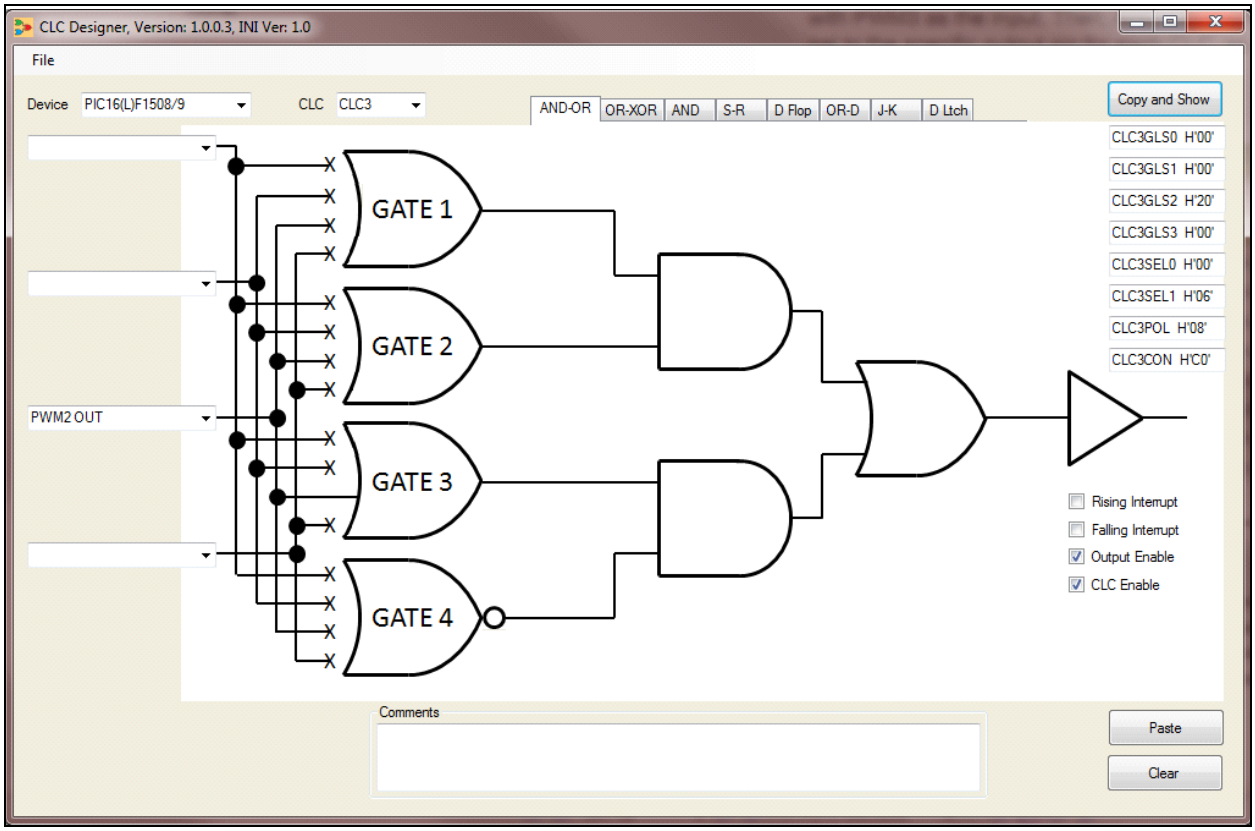
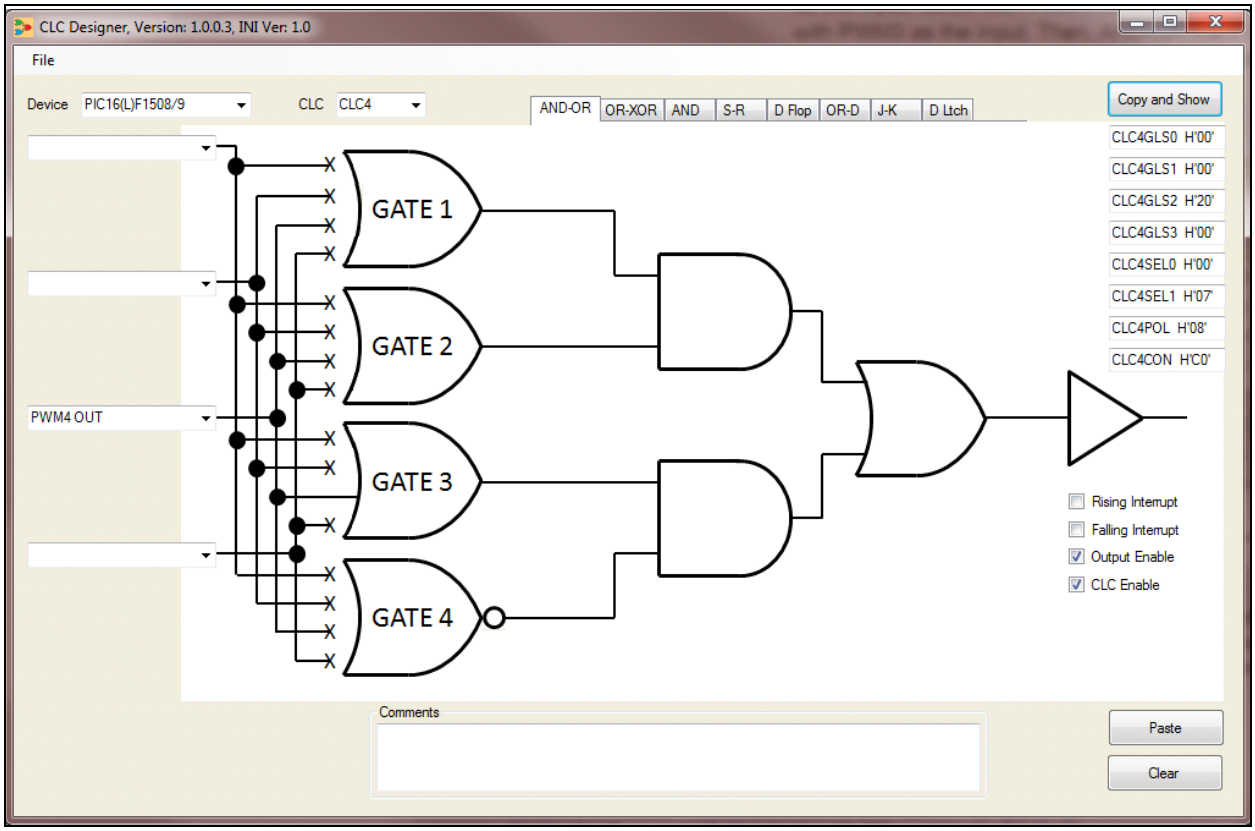


FIGURE 25: CLC4 CONFIGURATION FOR EXAMPLE 1



With the microcontroller configured this way, each PWM can be set up to output four different PWM signals. However, what if only one PWM signal is needed on up to four different output pins? See Example 2 below.

EXAMPLE 2

Again, using the PIC16F1508 device, only CLC2 will be set up with the output of PWM1 as the input, and all other CLC’s will be linked off of CLC2. This will put the output of PWM1 on four different output pins. See Figures 26, 27, 28 and 29.

FIGURE 26: CLC2 CONFIGURATION FOR EXAMPLE 2

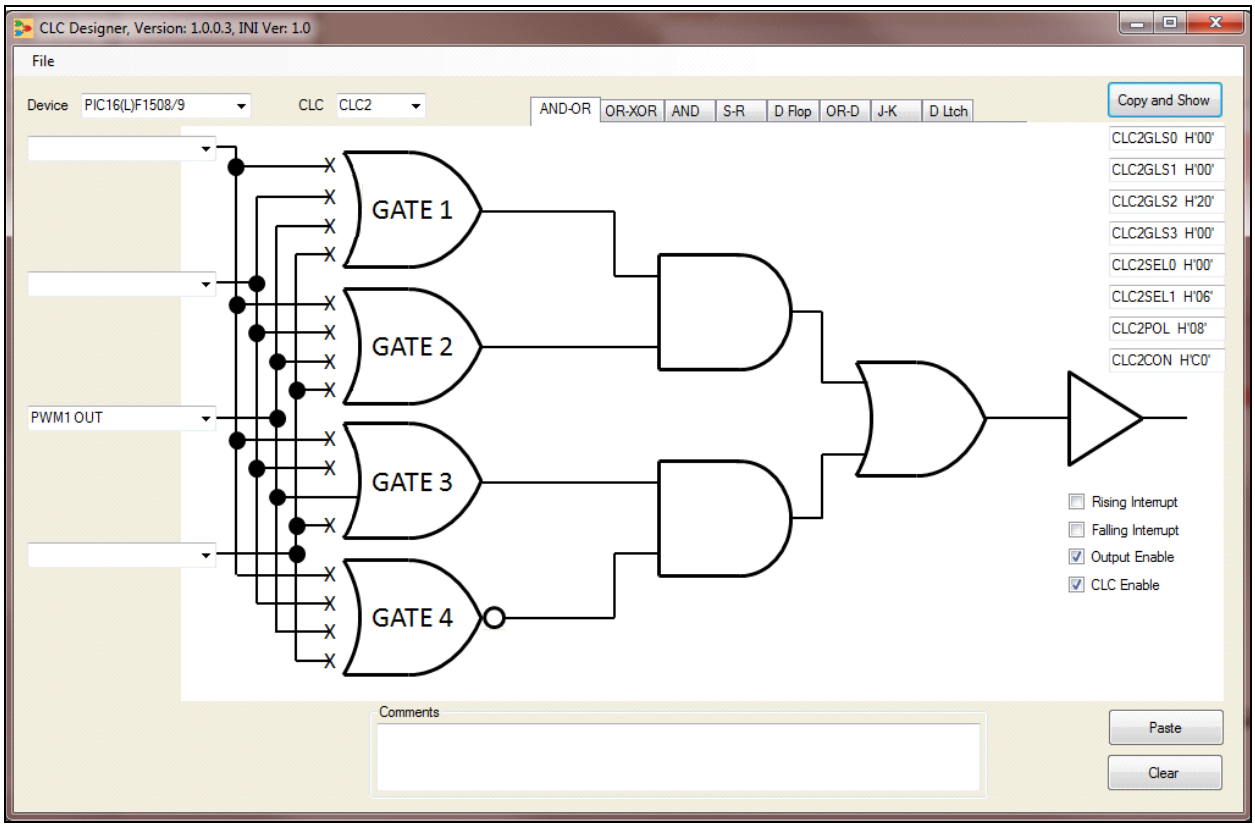


FIGURE 27: CLC1 CONFIGURATION FOR EXAMPLE 2

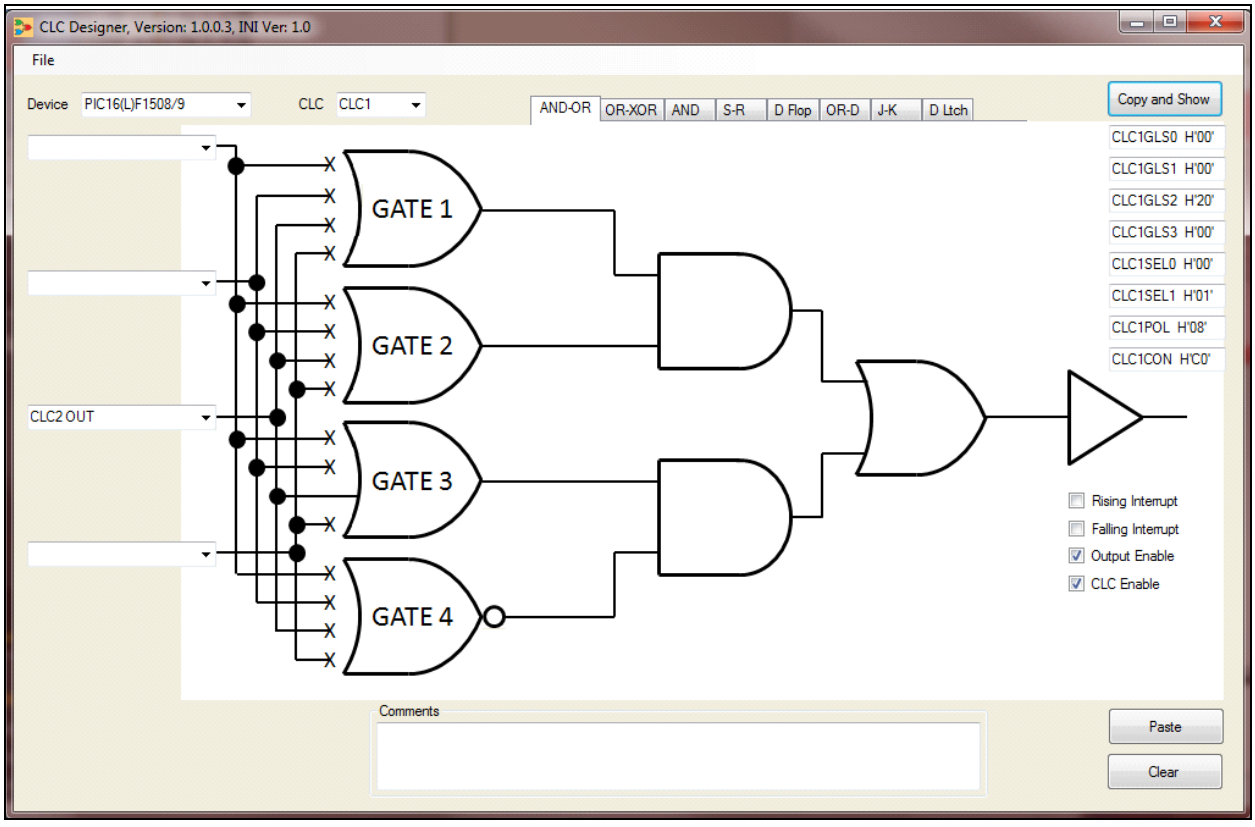


FIGURE 28: CLC3 CONFIGURATION FOR EXAMPLE 2

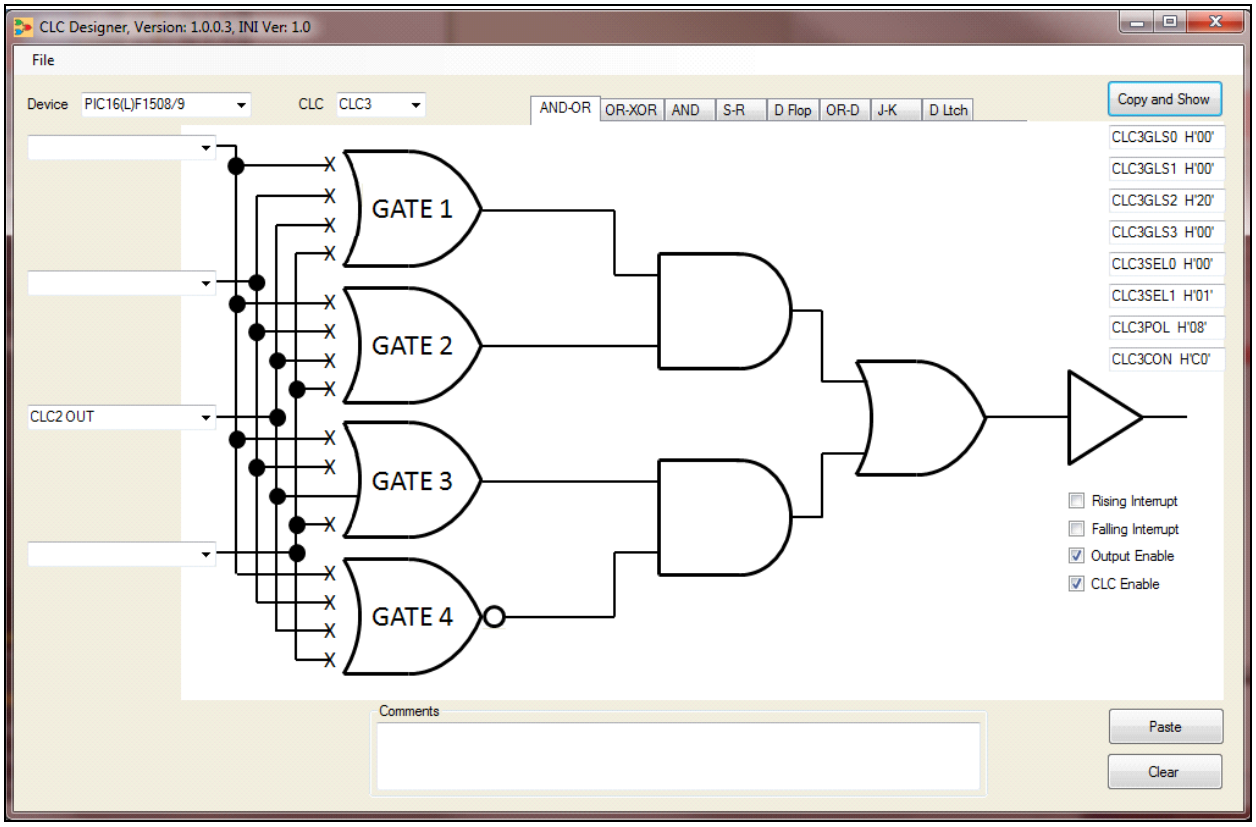
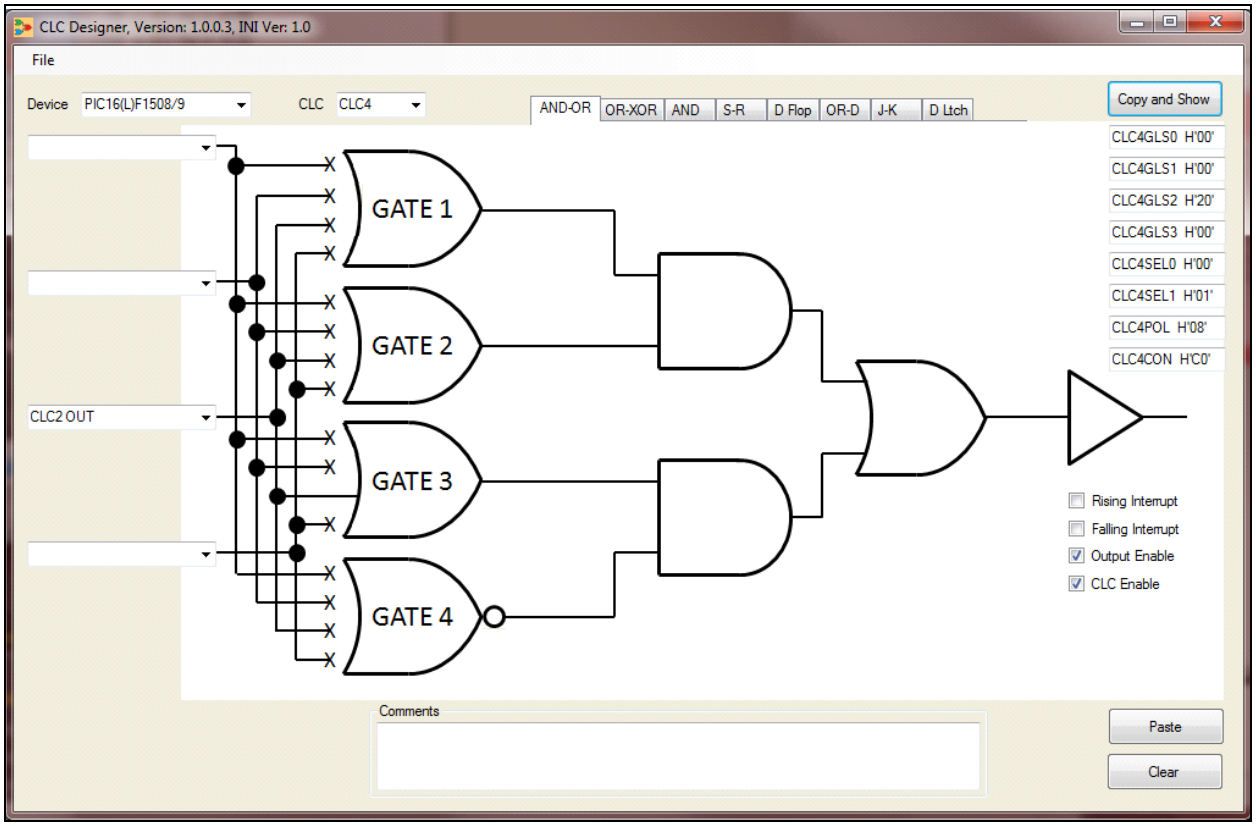


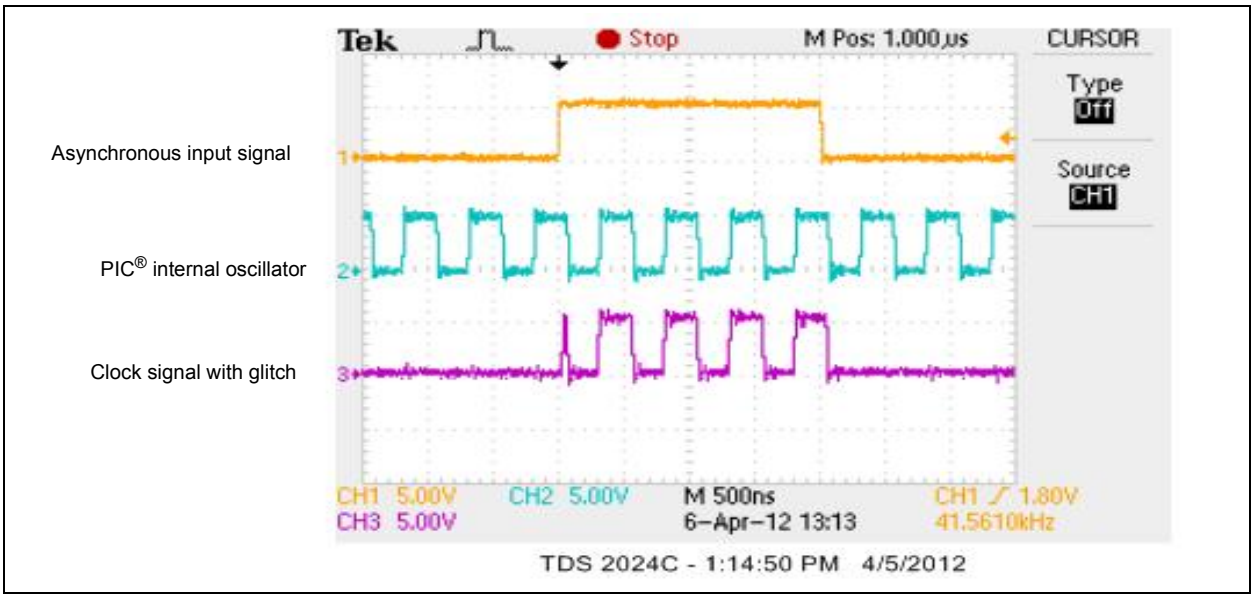
FIGURE 29: CLC4 CONFIGURATION FOR EXAMPLE 2



TIP 9: GLITCH-FREE CLOCK SIGNAL

A ‘glitch’ is a signal which does not remain active for a full clock period. If a signal with a glitch feeds the clock line of numerous latches, some of the latches may get updated, while others may not. This is clearly a situation that designers want to avoid.

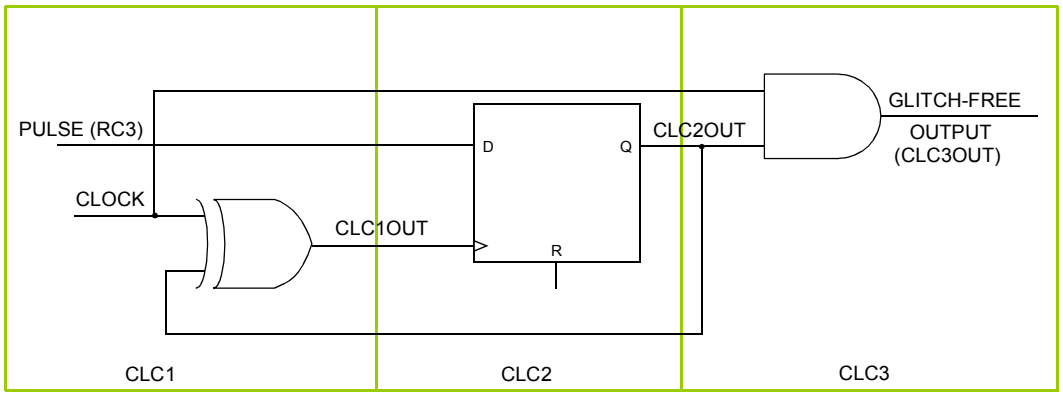
FIGURE 30: LOGICAL ‘AND’ OF ASYNCHRONOUS PULSE AND SYSTEM CLOCK



For this example, a PIC16F1509 was used because three of the four available internal CLC modules will be needed. The NCO will be used as a high speed counter to increment as long as an external pulse signal is high. This creates a high-resolution, long-duration counter, as the NCO counter is a 20-bit wide register. It will take approximately 16 instruction cycles (4 µs with 16 MHz clock) for the data to be read and the counter to reset, so it is necessary to have at least 4 µs of low time between pulses. A falling edge interrupt flag on CLC2 provides a signal that the pulse width measurement has been completed. CLC1 setup as an XOR gate will have the function of taking feedback from the D flip-flop (CLC2) and inverting the clock, so that it will trigger on the falling edge once the flip-flop has been set. CLC3 will use the ‘AND’ function to clock the NCO when the pulse is high. It will do this by creating a new signal that would only rise on the rising edge of the clock, and only fall on the falling edge of the clock. This new signal (CLC2OUT) is then AND’ed with the oscillator clock, thus insuring a glitch-free output signal. A simplified schematic for this is shown below (Figure 31).

Note:	For more detailed information, refer to application note AN1451 “*Glitch-Free Design Using the Configurable Logic Cell (CLC)*”.

FIGURE 31: CLC SETUP FOR GLITCH-FREE SIGNAL



TIP 10: DELAY BLOCK/DEBOUNCER

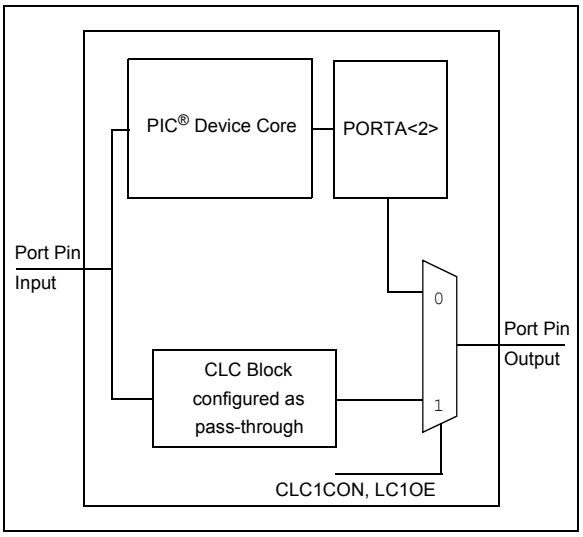
In this example, the Configurable Logic Cell (CLC) is being used to implement a delay block/debouncer. The delay can be set between 2µs and 193 µs, which is used effectively as a noise discriminator, or for switch debouncing.

When used as a delay block, the application can be used to fix low-level timing issues on signals. When used as a debouncer, it can debounce signals from a mechanical switch, so that a clean signal can feed other circuitry.

The Configurable Logic Cell peripheral is used to produce fast switching on the output (if desired). If the same application were written using port logic only, there would be multiple instruction cycles before the output would change in response to an input. Thus, in using the CLC, the signal can be routed directly and only have propagation and gate delay between the input and output signals.

With the CLC block configured as a pass-through, it is possible to quickly route signals to the output when no delay is desired, and the PIC® device core (port function) will create edge delays when desired. The MUX (CLC1CON, LC1OE) selects whether the pin is driven by the CLC or by the port logic (Figure 32).

FIGURE 32: BLOCK DIAGRAM



Note: For detailed information, refer to application note AN1450 “*Delay Block/ Debouncer*”.

RESOURCES

[1] *Configurable Logic Cell (CLC) Configuration Tool User’s Guide*, DS41597 at www.microchip.com.

[2] Configurable Logic Cell (CLC) Configuration Tool GUI software at www.microchip.com.

[3] Device data sheet for the specific device being used, at www.microchip.com.

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

Printed on recycled paper.

ISBN: 9781620766033

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Osaka
Tel: 81-66-152-7160
Fax: 81-66-152-9310

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/11