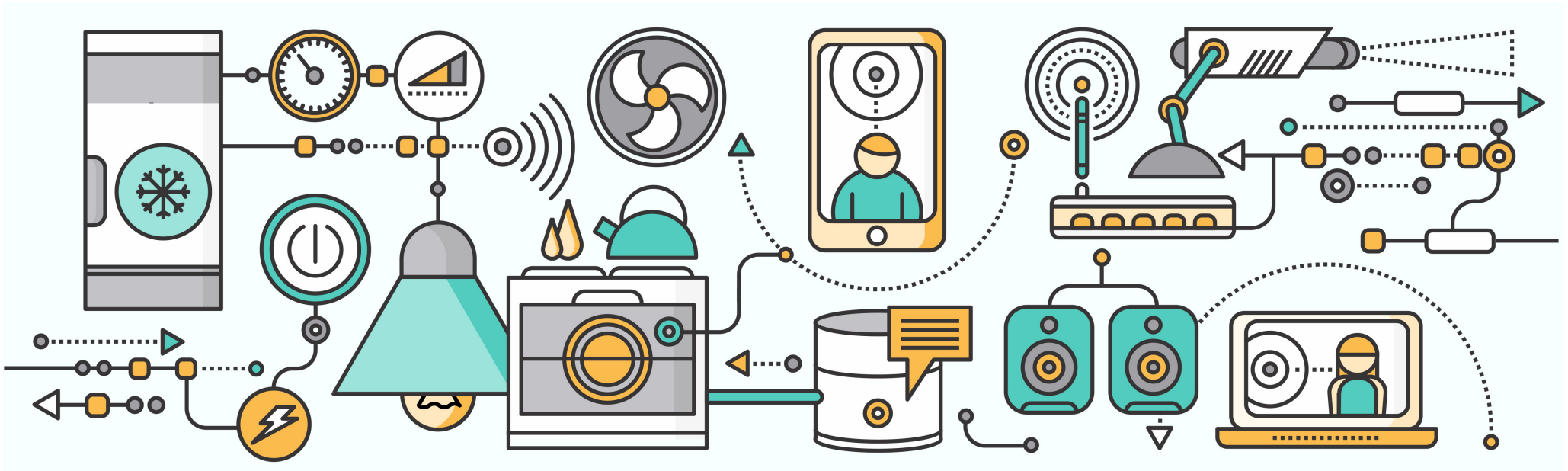
ENME 441
Mechatronics and the Internet of Things



Shift Registers

Input shift register: parallel input, serial output

- Load multiple voltages from different input pins in parallel
- Shift each input voltage to a serial output to be read by the Pi
- Common 8-bit input shift register: TI SN74HC165

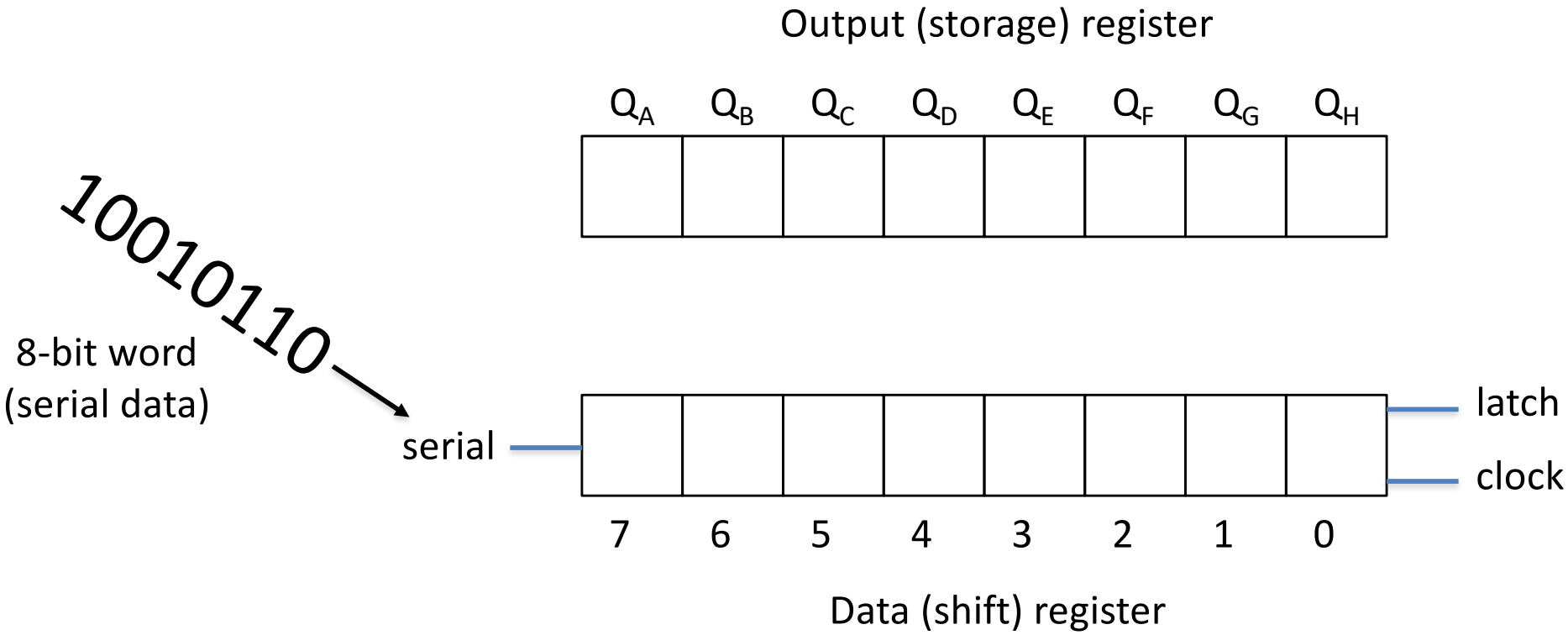
Output shift register: serial input, parallel output

- Send multiple voltages in series from the Pi
- Shift each voltage to a different output pin for parallel outputs
- Common 8-bit output shift register: TI SN74HC595

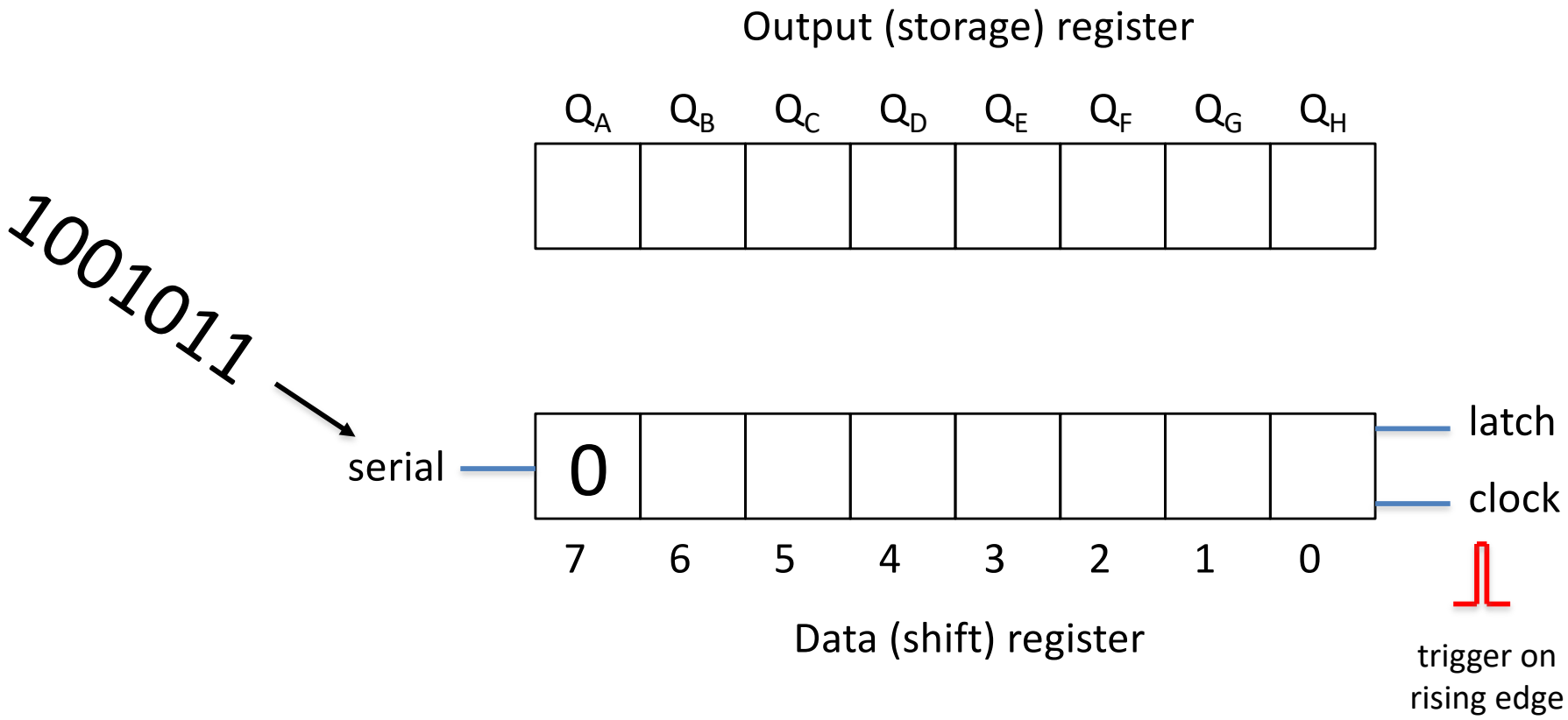
Latching Output Shift Registers

- Shift register = serial-to-parallel converter
  - binary sequence of serial input data → parallel output data block.
  - Allows multiple outputs to be controlled from a single GPIO pin.
- Latching = output values are loaded sequentially and applied to all outputs simultaneously on a latch signal
- Shift registers isolate GPIO pins from high-current outputs
- Serial, Clock, Latch, and Output pins:
  - Serial → serial input data
  - Clock → shift register data one step, and place current input in 1st register
  - Latch → move all register data to parallel output lines simultaneously
  - Output → set of parallel output lines

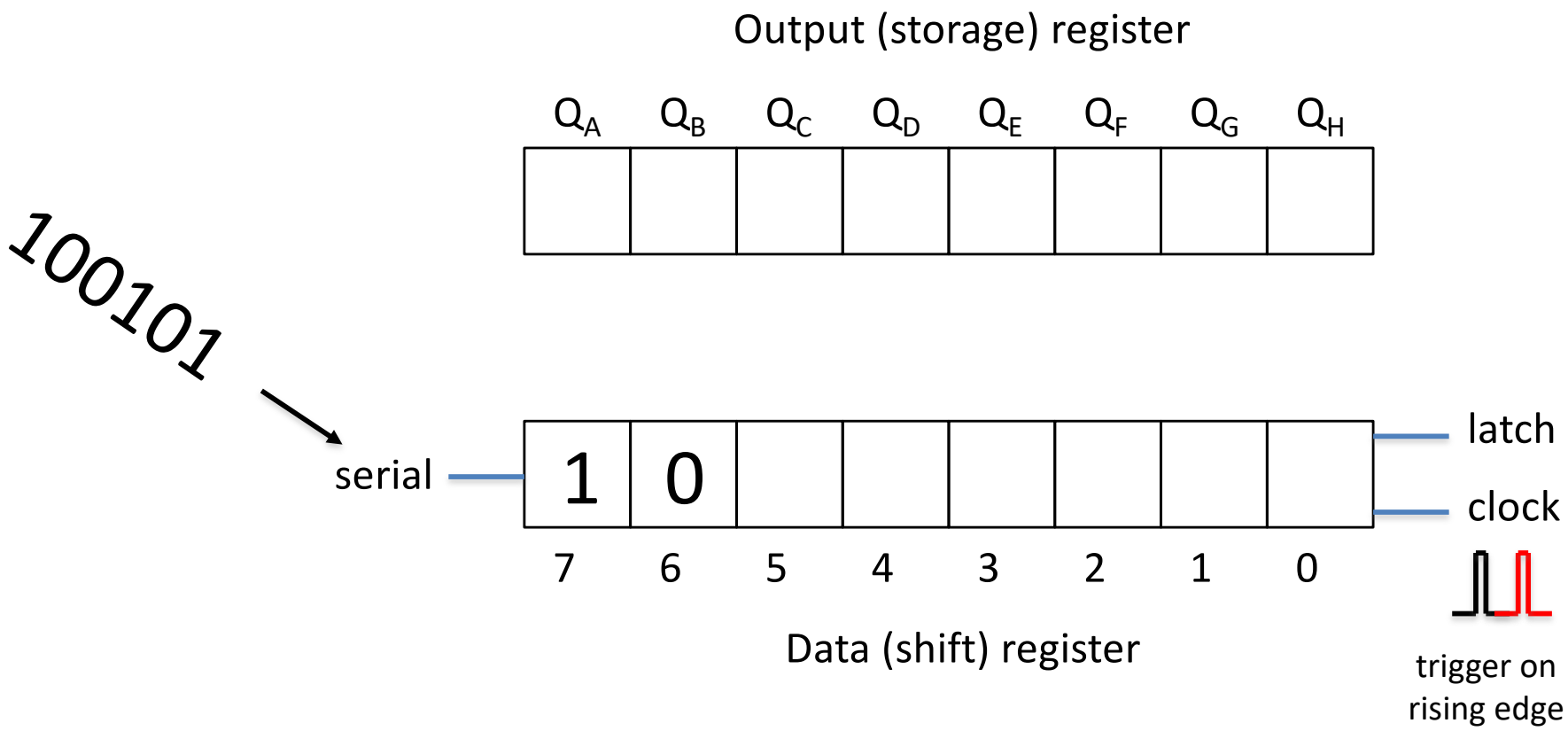
Output Shift Register Operation



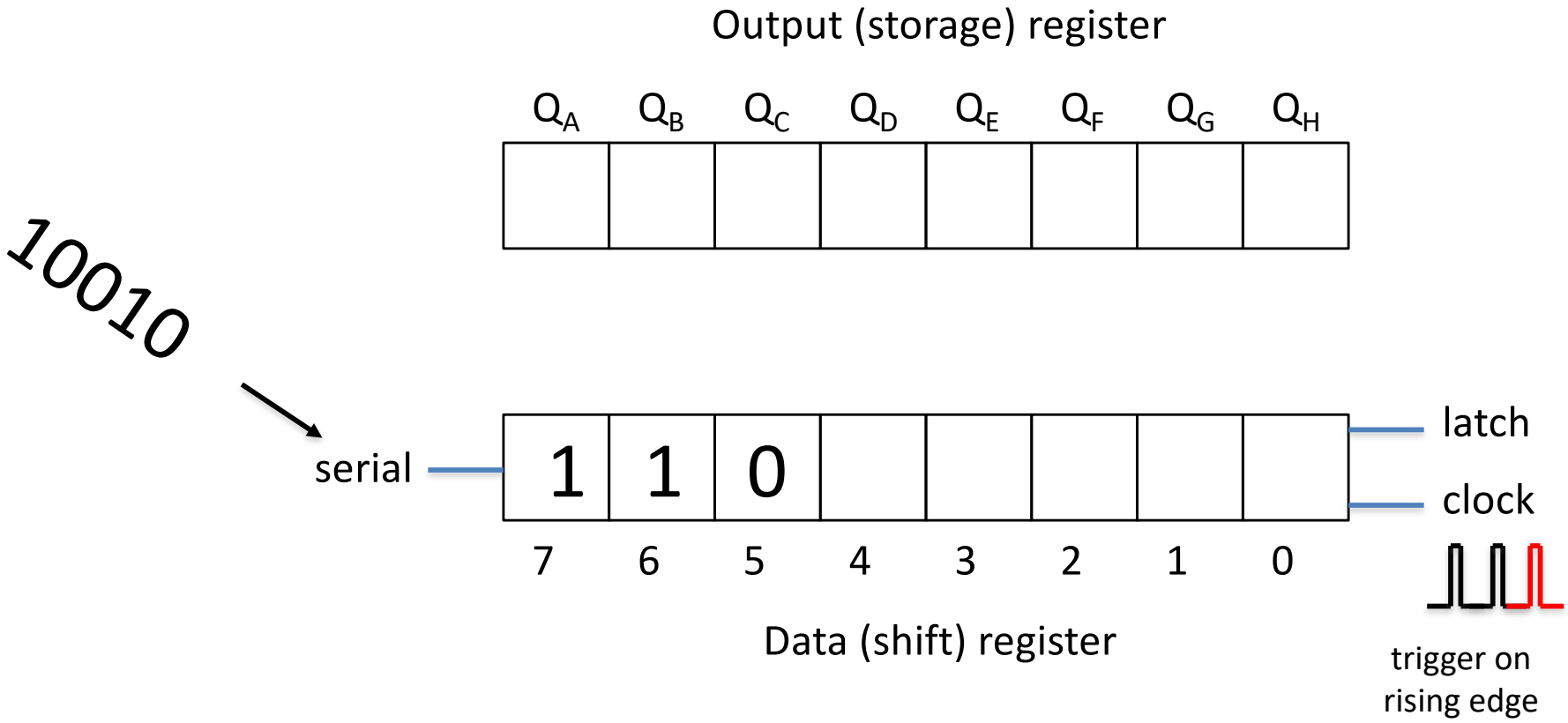
Output Shift Register Operation



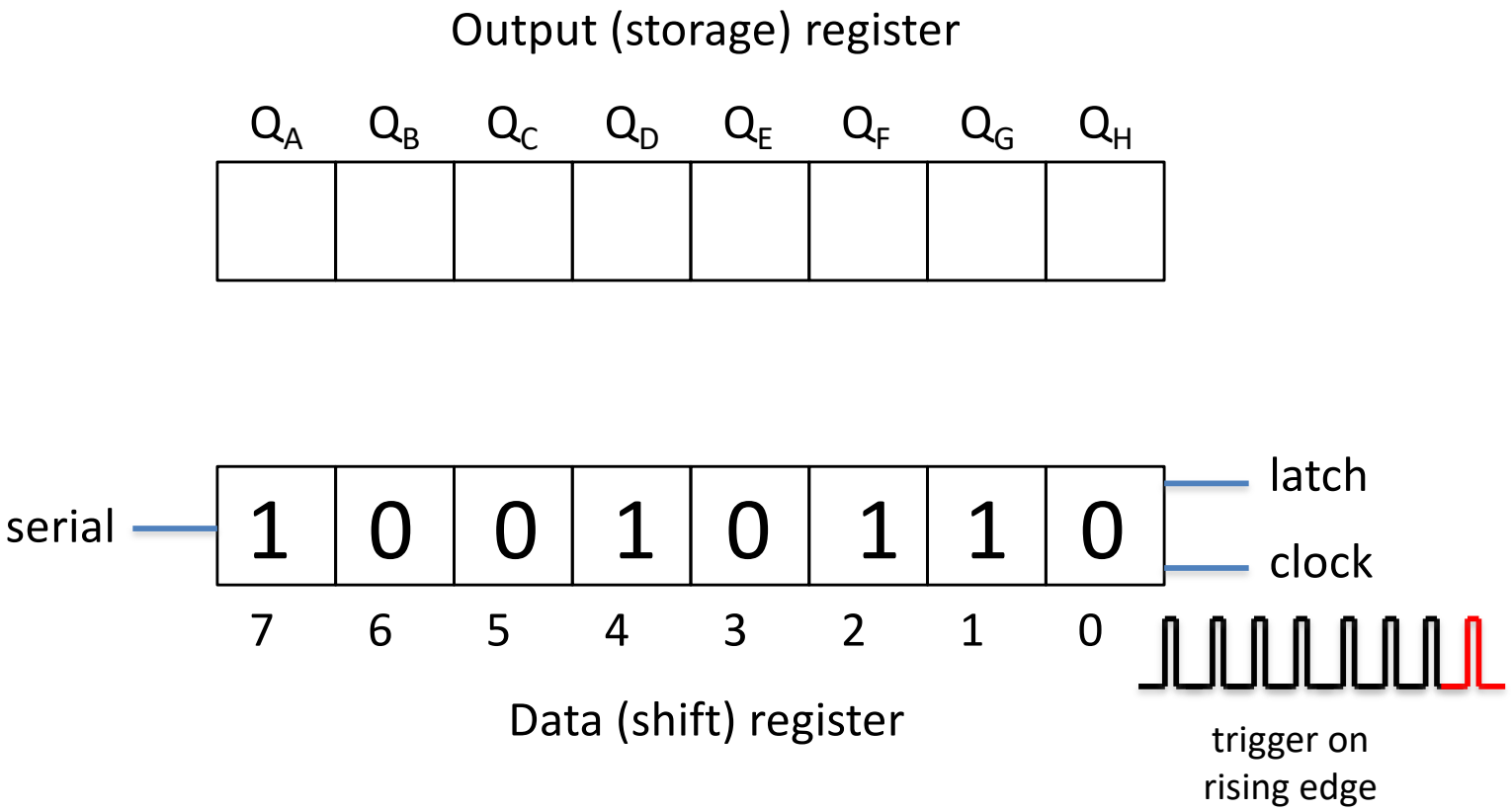
Output Shift Register Operation



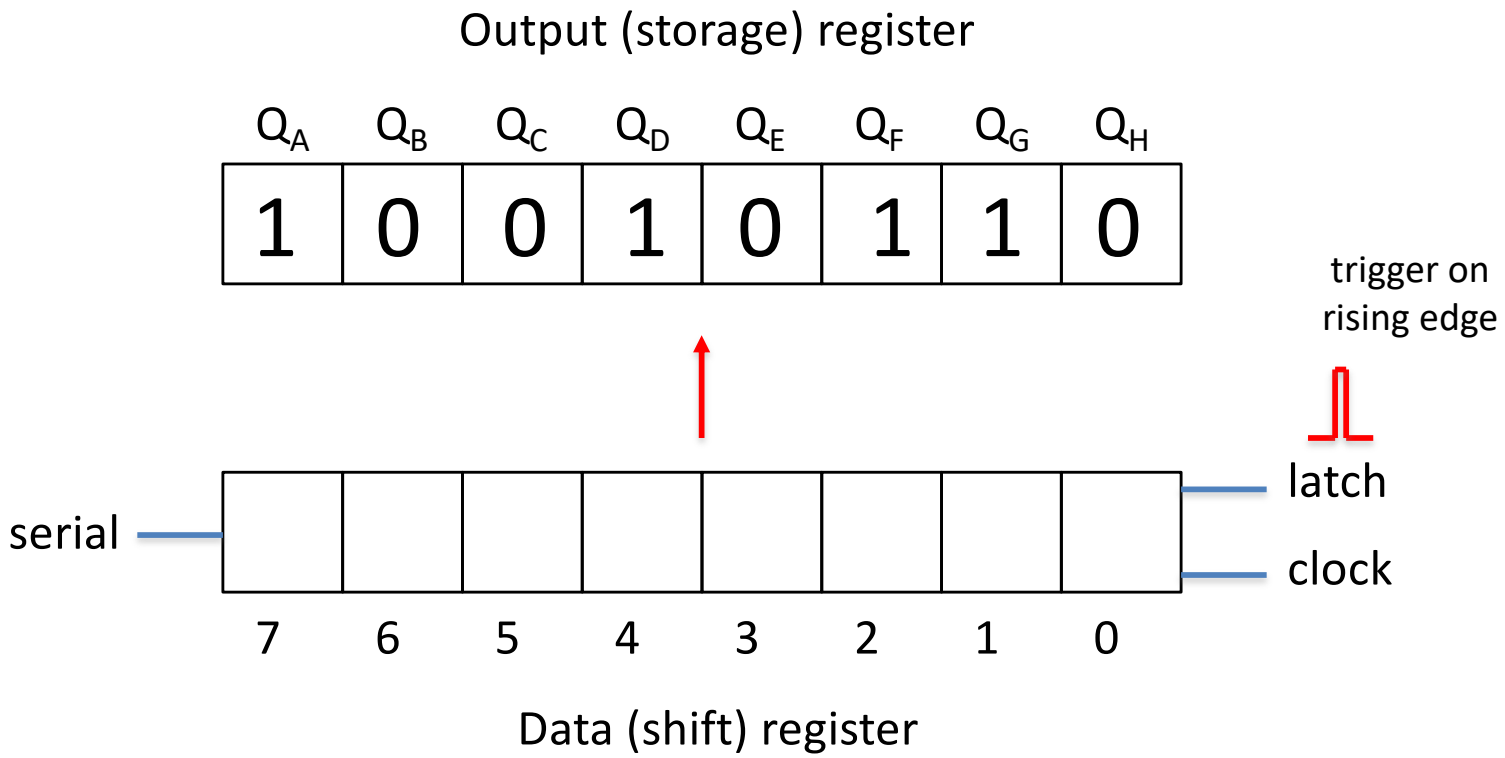
Output Shift Register Operation



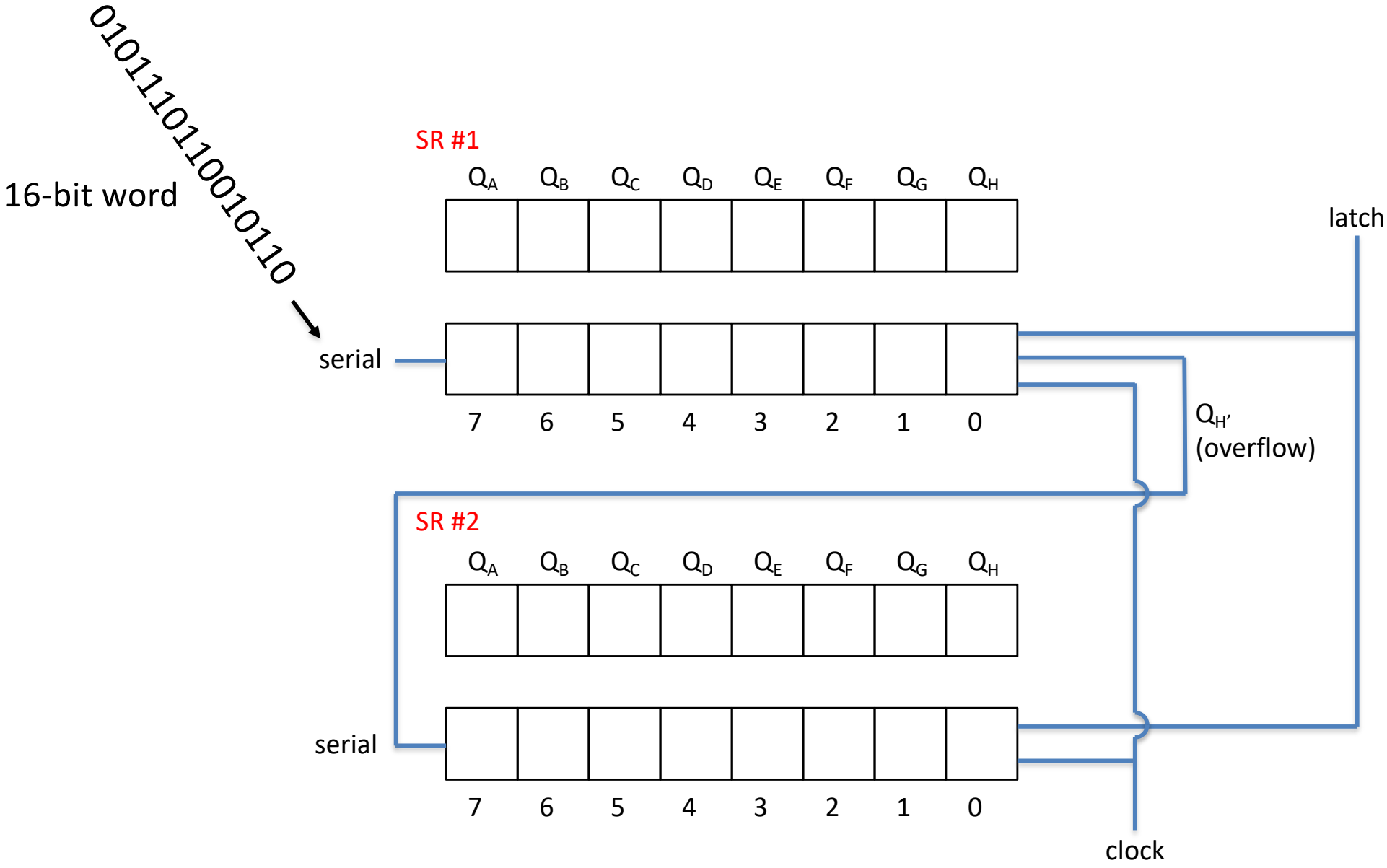
Output Shift Register Operation



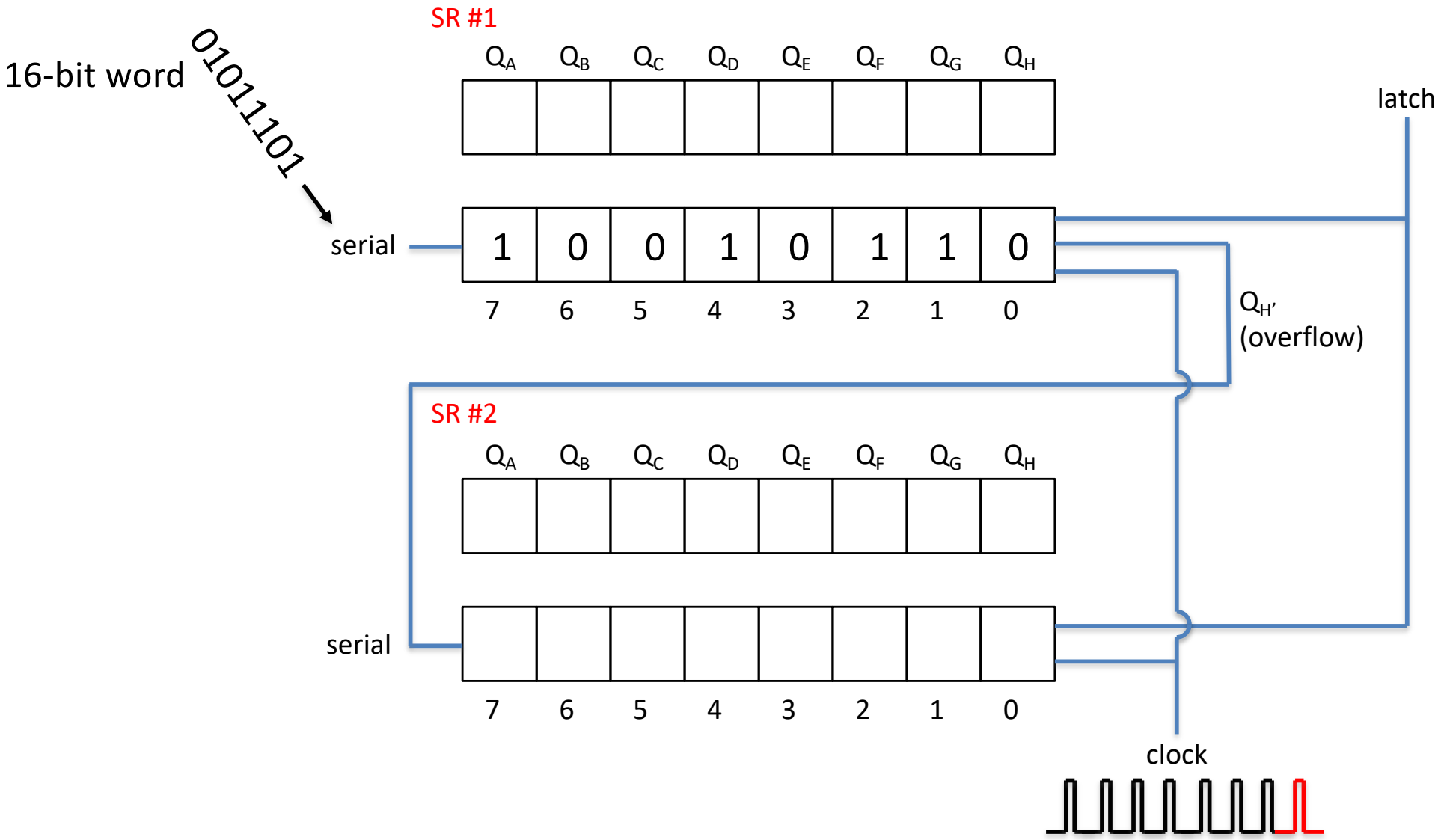
Output Shift Register Operation



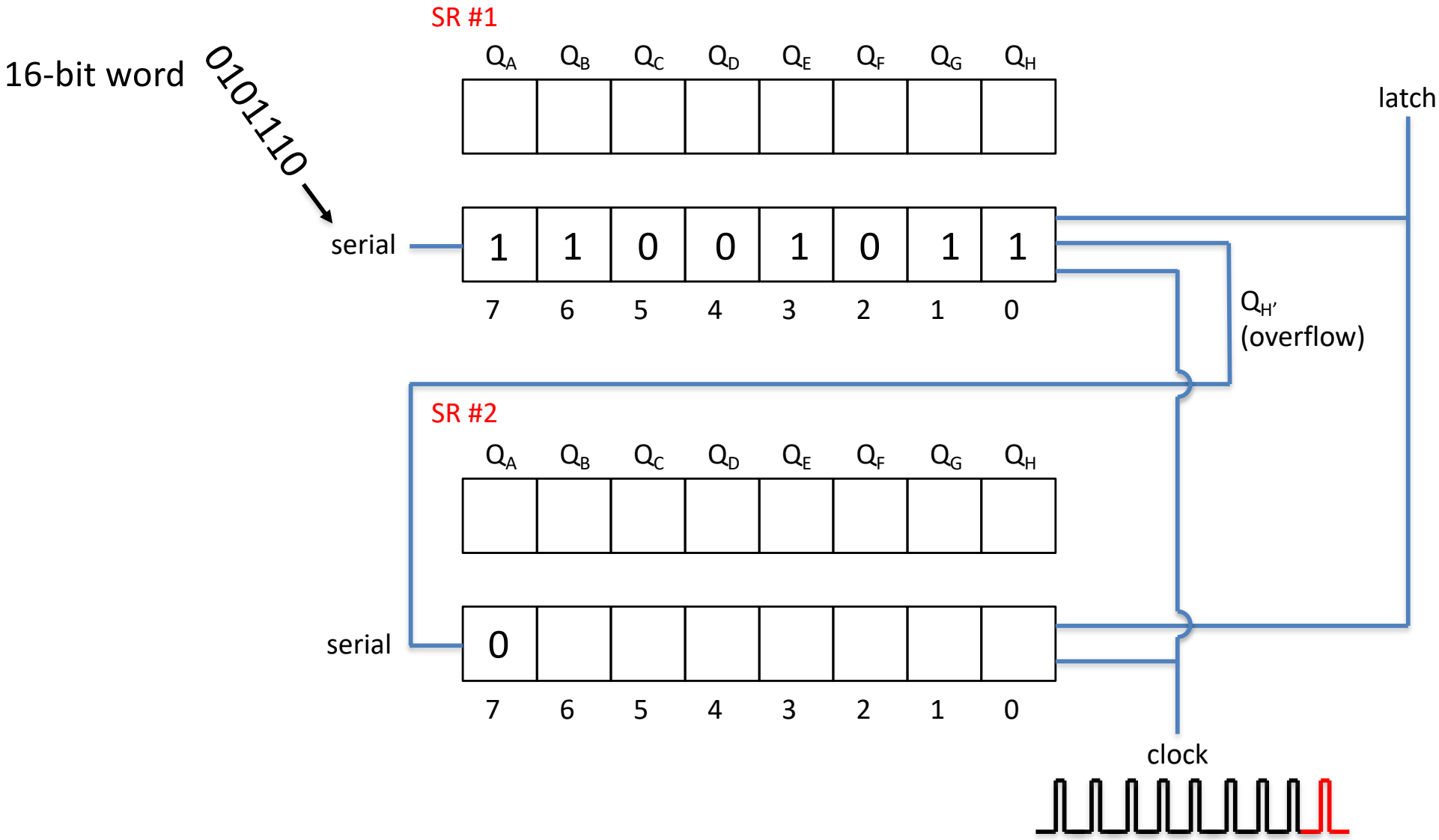
Cascaded Shift Registers



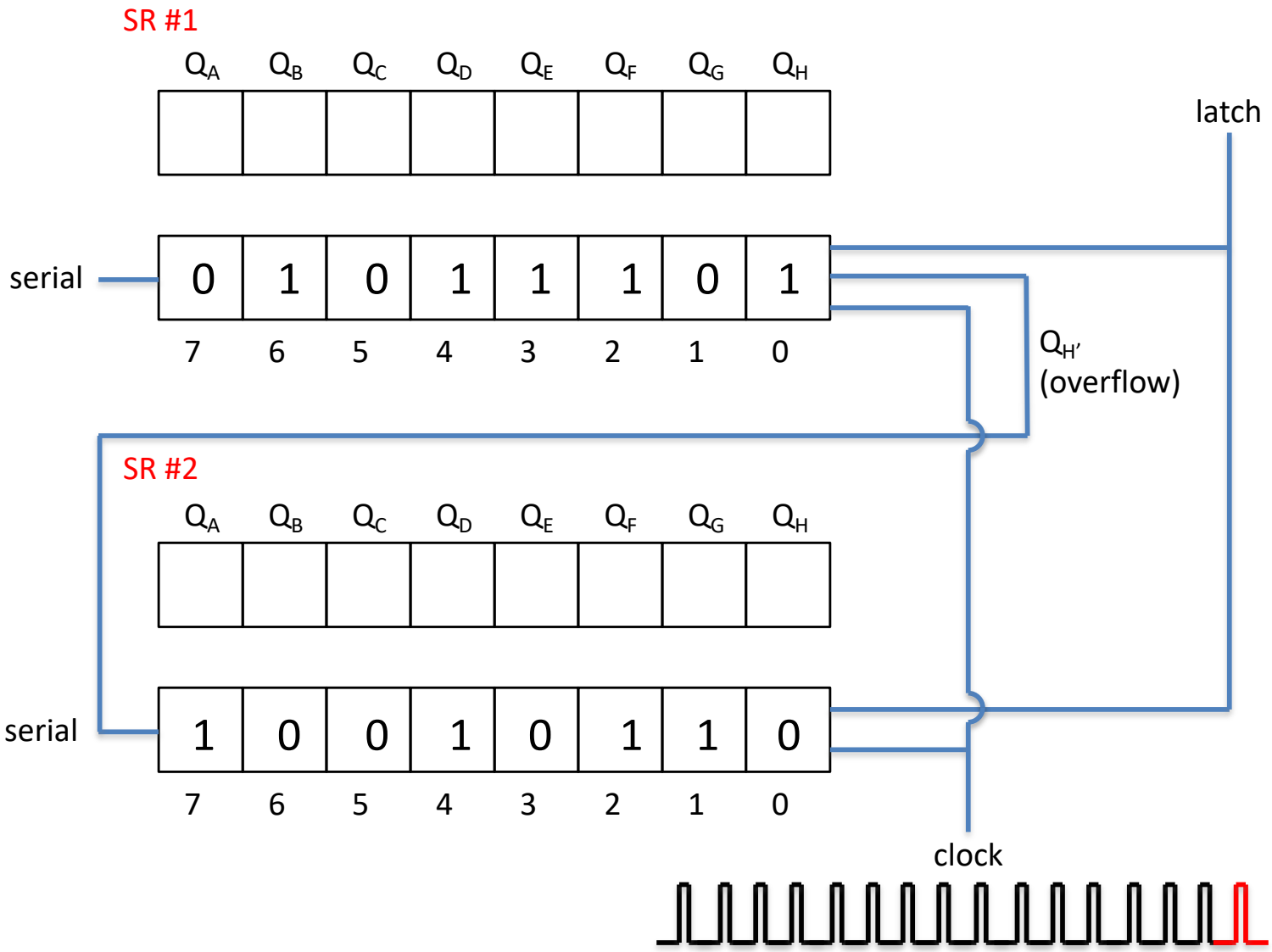
Cascaded Shift Registers



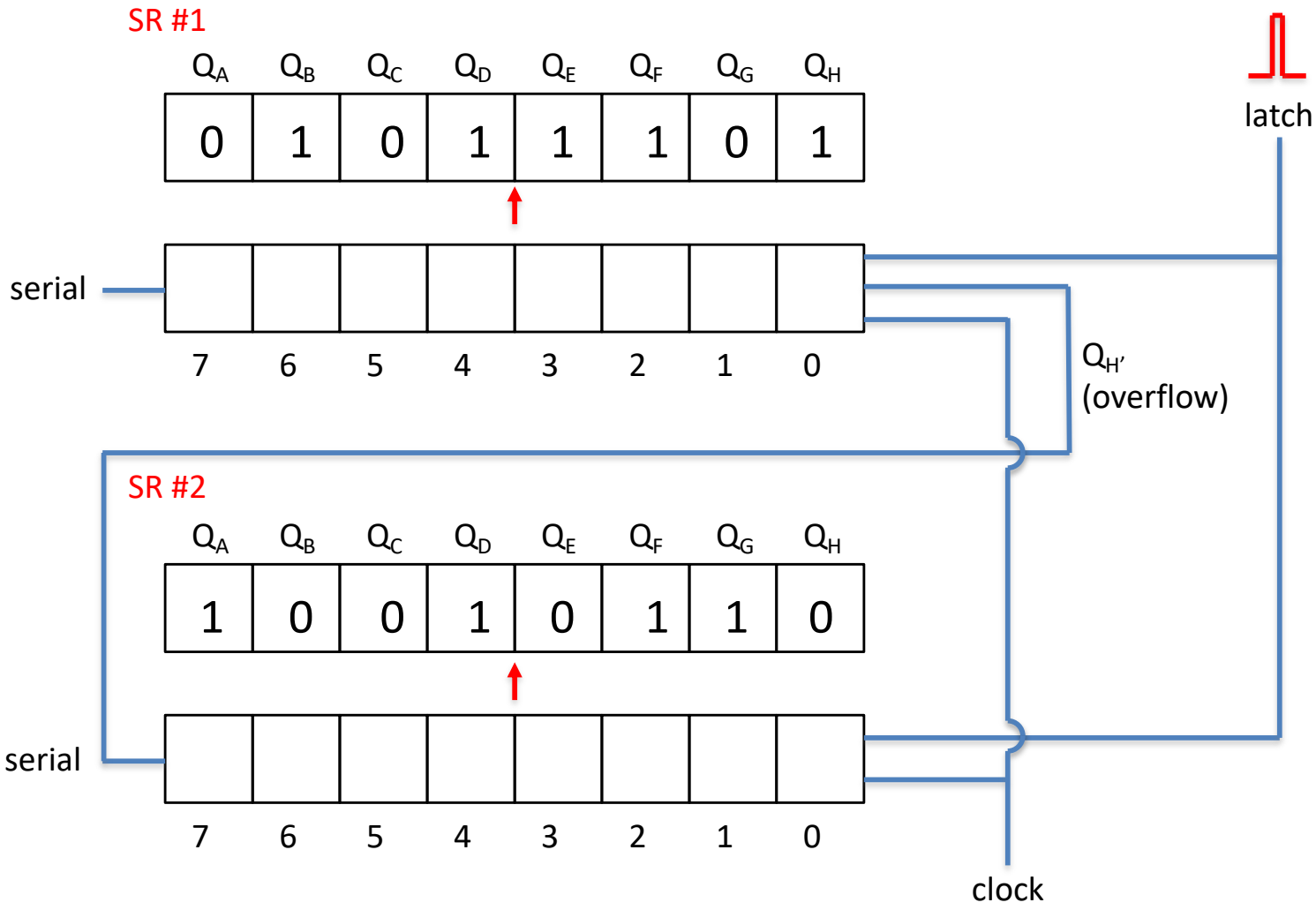
Cascaded Shift Registers



Cascaded Shift Registers



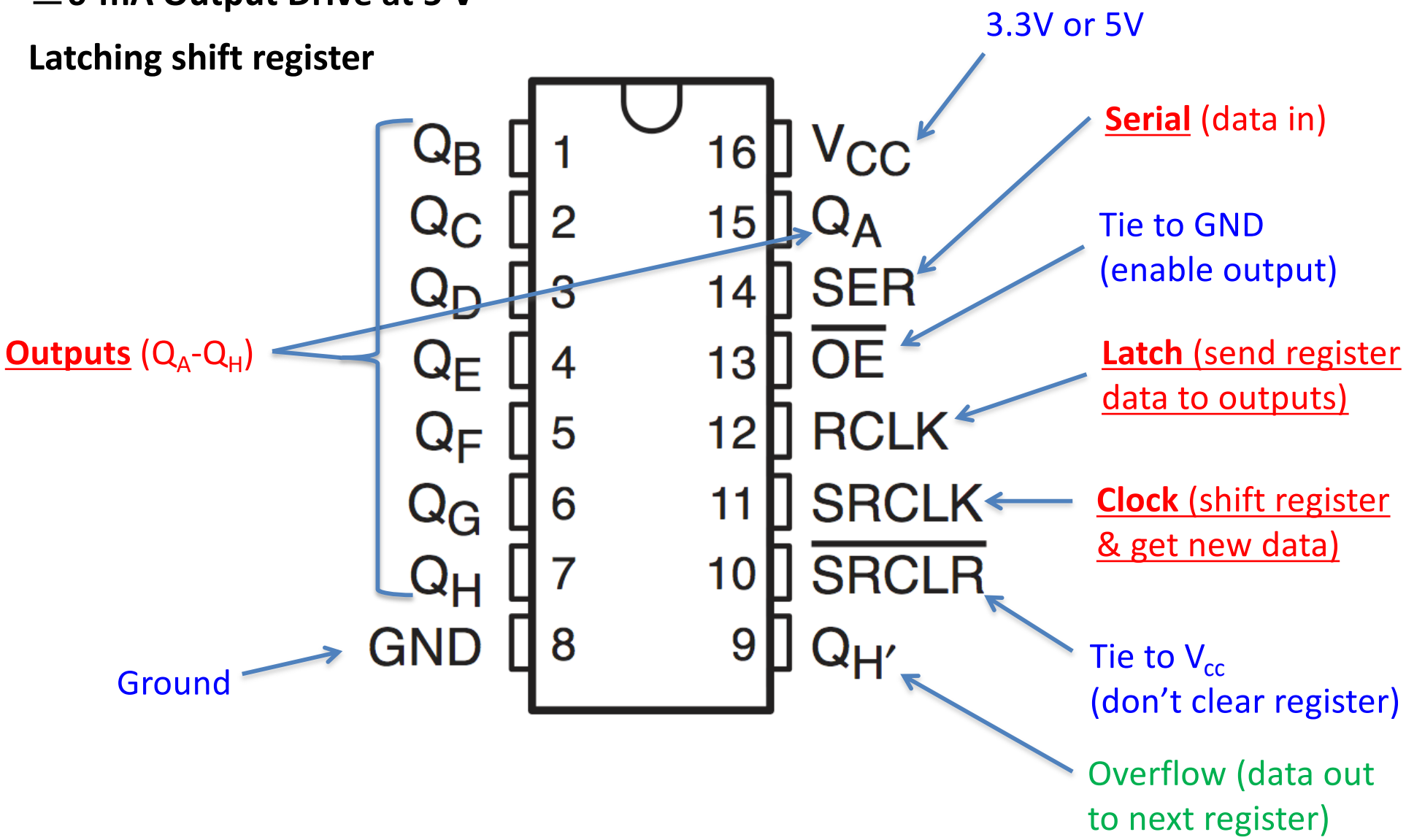
Cascaded Shift Registers



SN74HC595 Pinout (DIP-16 package)

±6-mA Output Drive at 5 V

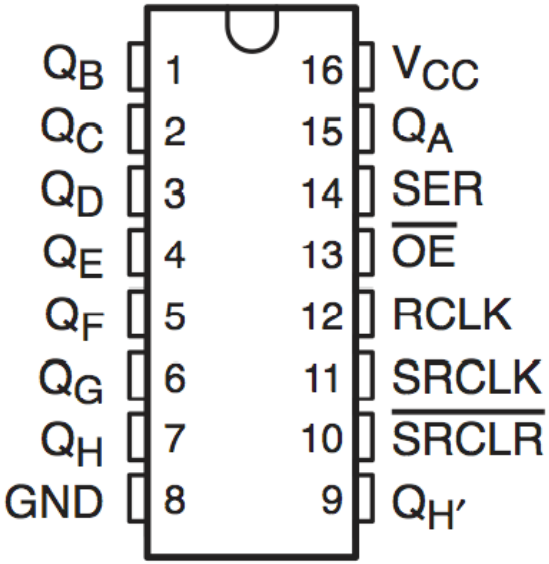
Latching shift register



SN74HC595 Datasheet

OE = output-enable
SRCLR = shift register clear

SER = serial input (binary data)
SRCLK = shift register clock (“clock”)
RCLK = storage register clock (“latch”)



INPUTS					FUNCTION
SER	SRCLK	$\overline{SRCLR}$	RCLK	$\overline{OE}$	
X	X	X	X	H	Outputs Q_A–Q_H are disabled.
X	X	X	X	L	Outputs Q_A–Q_H are enabled.
X	X	L	X	X	Shift register is cleared.
L	↑	H	X	X	First stage of the shift register goes low. Other stages store the data of previous stage, respectively.
H	↑	H	X	X	First stage of the shift register goes high. Other stages store the data of previous stage, respectively.
X	X	X	↑	X	Shift-register data is stored in the storage register.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

dataPin, latchPin, clockPin = 23, 24, 25

GPIO.setup(dataPin, GPIO.OUT)
GPIO.setup(latchPin, GPIO.OUT, initial=0)  # start latch & clock low
GPIO.setup(clockPin, GPIO.OUT, initial=0)

pattern = 0b01100110            # pattern to display

for i in range(8):
  GPIO.output(dataPin, pattern & (1<<i))
  GPIO.output(clockPin,1)       # ping the clock pin to shift register data
  time.sleep(0)
  GPIO.output(clockPin,0)

GPIO.output(latchPin, 1)        # ping the latch pin to send register to output
time.sleep(0)
GPIO.output(latchPin, 0)

try:
  while 1: pass
except:
  GPIO.cleanup()
```

Download `shift_reg_initial.py` to your Pi by updating from the ENME441 github repository:

```
cd ~/enme441-pi
git pull
```

- ***Repeating code...move to a function called ping()***

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

dataPin, latchPin, clockPin = 23, 24, 25

GPIO.setup(dataPin, GPIO.OUT)
GPIO.setup(latchPin, GPIO.OUT, initial=0)  # start latch & clock low
GPIO.setup(clockPin, GPIO.OUT, initial=0)

pattern = 0b01100110            # pattern to display

for i in range(8):
  GPIO.output(dataPin, pattern & (1<<i))
  GPIO.output(clockPin,1)       # ping the clock pin to shift register data
  time.sleep(0)
  GPIO.output(clockPin,0)

GPIO.output(latchPin, 1)        # ping the latch pin to send register to output
time.sleep(0)
GPIO.output(latchPin, 0)

try:
  while 1: pass
except:
  GPIO.cleanup()
```

Download `shift_reg_initial.py` to your Pi by updating from the ENME441 github repository:

```
cd ~/enme441-pi
git pull
```

- ***Repeating code...move to a function called ping()***

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

dataPin, latchPin, clockPin = 23, 24, 25

GPIO.setup(dataPin, GPIO.OUT)
GPIO.setup(latchPin, GPIO.OUT, initial=0)  # start latch & clock low
GPIO.setup(clockPin, GPIO.OUT, initial=0)

pattern = 0b01100110  # pattern to display

def ping(p):          # ping the clock or latch pin
  GPIO.output(p,1)
  time.sleep(0)
  GPIO.output(p,0)

for i in range(8):
  GPIO.output(dataPin, pattern & (1<<i))
  ping(clockPin)      # ping the clock pin to shift register data

ping(latchPin)        # ping the latch pin to send register to output

try:
  while 1: pass
except:
  GPIO.cleanup()
```

- ***Move byte shifting to a new function***

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

dataPin, latchPin, clockPin = 23, 24, 25

GPIO.setup(dataPin, GPIO.OUT)
GPIO.setup(latchPin, GPIO.OUT, initial=0)  # start latch & clock low
GPIO.setup(clockPin, GPIO.OUT, initial=0)

pattern = 0b01100110  # pattern to display

def ping(p):          # ping the clock or latch pin
  GPIO.output(p,1)
  time.sleep(0)
  GPIO.output(p,0)

for i in range(8):
  GPIO.output(dataPin, pattern & (1<<i))
  ping(clockPin)      # ping the clock pin to shift register data

ping(latchPin)        # ping the latch pin to send register to output

try:
  while 1: pass
except:
  GPIO.cleanup()
```

- ***Move byte shifting to a new function***

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

dataPin, latchPin, clockPin = 23, 24, 25

GPIO.setup(dataPin, GPIO.OUT)
GPIO.setup(latchPin, GPIO.OUT, initial=0)  # start latch & clock low
GPIO.setup(clockPin, GPIO.OUT, initial=0)

pattern = 0b01100110  # pattern to display

def ping(p):          # ping the clock or latch pin
  GPIO.output(p,1)
  time.sleep(0)
  GPIO.output(p,0)

def shiftByte(b):       # send a byte of data to the output
  for i in range(8):
    GPIO.output(dataPin, b & (1<<i))
    ping(clockPin)      # add bit to register
  ping(latchPin)        # send register to output

try:
  shiftByte(0b01100110)   # test out the new function
  while 1: pass
except:
  GPIO.cleanup()
```

- ***Display an 8-bit binary counter***

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

dataPin, latchPin, clockPin = 23, 24, 25

GPIO.setup(dataPin, GPIO.OUT)
GPIO.setup(latchPin, GPIO.OUT, initial=0)  # start latch & clock low
GPIO.setup(clockPin, GPIO.OUT, initial=0)

pattern = 0b01100110  # pattern to display

def ping(p):          # ping the clock or latch pin
  GPIO.output(p,1)
  time.sleep(0)
  GPIO.output(p,0)

def shiftByte(b):       # send a byte of data to the output
  for i in range(8):
    GPIO.output(dataPin, b & (1<<i))
    ping(clockPin)      # add bit to register
  ping(latchPin)        # send register to output

try:
  shiftByte(0b01100110)   # test out the new function
  while 1: pass
except:
  GPIO.cleanup()
```

- ***Display an 8-bit binary counter***

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

dataPin, latchPin, clockPin = 23, 24, 25

GPIO.setup(dataPin, GPIO.OUT)
GPIO.setup(latchPin, GPIO.OUT, initial=0)  # start latch & clock low
GPIO.setup(clockPin, GPIO.OUT, initial=0)

pattern = 0b01100110  # pattern to display

def ping(p):          # ping the clock or latch pin
  GPIO.output(p,1)
  time.sleep(0)
  GPIO.output(p,0)

def shiftByte(b):       # send a byte of data to the output
  for i in range(8):
    GPIO.output(dataPin, b & (1<<i))
    ping(clockPin)      # add bit to register
  ping(latchPin)        # send register to output

try:
  while 1:
    for i in range(2**8):
      shiftByte(i)
      time.sleep(0.5)
except:
  GPIO.cleanup()
```

- ***Move to a new Shifter class***

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

dataPin, latchPin, clockPin = 23, 24, 25

GPIO.setup(dataPin, GPIO.OUT)
GPIO.setup(latchPin, GPIO.OUT, initial=0)  # start latch & clock low
GPIO.setup(clockPin, GPIO.OUT, initial=0)

pattern = 0b01100110  # pattern to display

def ping(p):          # ping the clock or latch pin
  GPIO.output(p,1)
  time.sleep(0)
  GPIO.output(p,0)

def shiftByte(b):       # send a byte of data to the output
  for i in range(8):
    GPIO.output(dataPin, b & (1<<i))
    ping(clockPin)      # add bit to register
  ping(latchPin)        # send register to output

try:
  while 1:
    for i in range(2**8):
      shiftByte(i)
      time.sleep(0.5)
except:
  GPIO.cleanup()
```