

# **PAT 451/551**

# **INTERACTIVE**

# **MEDIA**

# **DESIGN I**

**MULTIPLEXING**

# MULTIPLEXING

**Problem:** We want to connect more than 6 analog inputs to the Arduino.

**Solution:** Use a multiplexer. It allows multiple analog sources to be connected to a single analog input. You use digital outputs from the Arduino to switch between the analog sources.

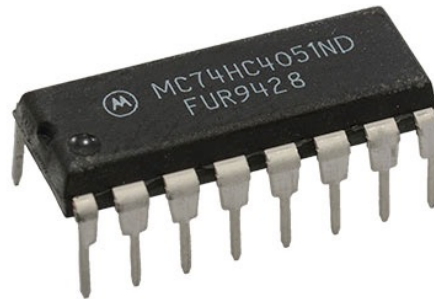
# MULTIPLEXERS

Typically an “Integrated Circuit” (IC), aka, a “chip.”

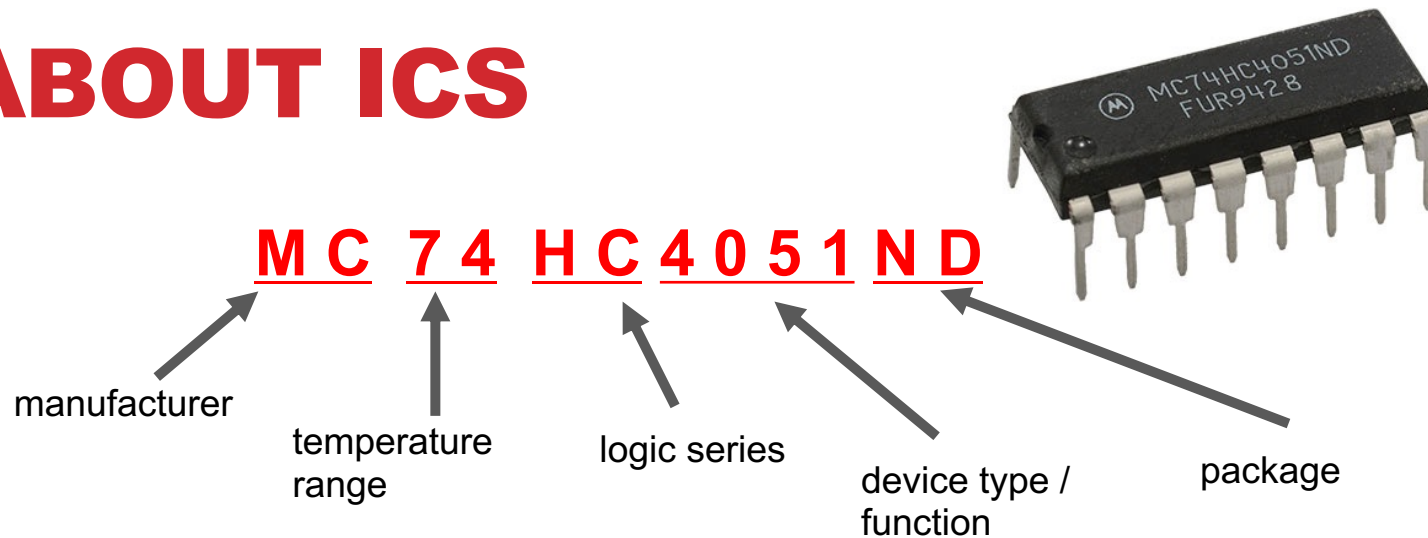
A very common mux chip you’ll encounter is a “**4051**”.

ICs are identified by an alphanumeric code, where the central numbers indicate the functional/logical operation of the chip.

This 4051 IC is “MC74HC4051ND”

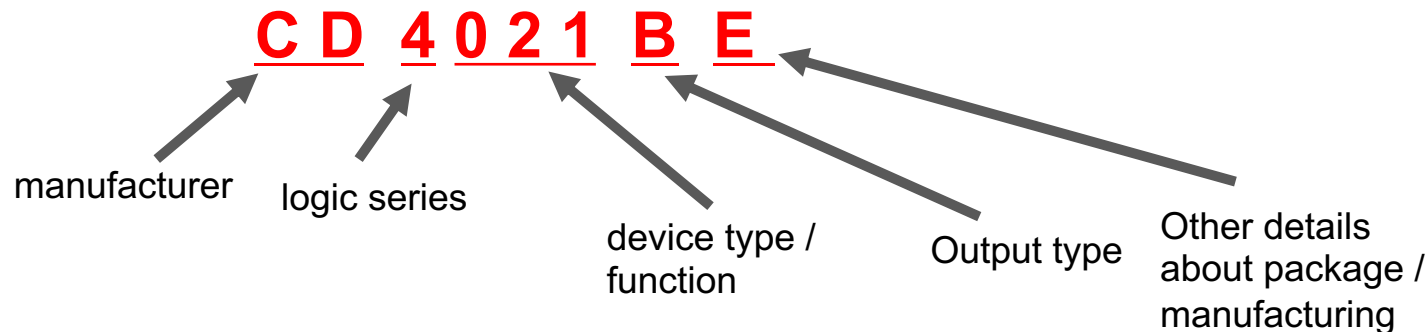


# ABOUT ICS



- **Manufacturer:** normally two letters. “MC” is Motorola. “SN” is Texas Instruments, etc. Not important.
- **Temperature Range:** 74 is “commercial” (0-70° C), 54 is “military” spec (-55-125° C).
- **Logic Series:** Specifies electrical properties, e.g., the supply voltage range, current range, timing, power consumption, and impedance qualities of input/output. If you are purchasing components, try to opt for “HC.” Other options are HCT, LS.
- **Device Type:** What the chip actually *does*. A 4051 is an 8-to-1 multiplexer.
- **Package:** physical properties of the chip, most important is whether it is surface mount or DIP (dual-inline package). DIP will fit in our breadboard!

# 4000 SERIES ICS (JUST FYI)



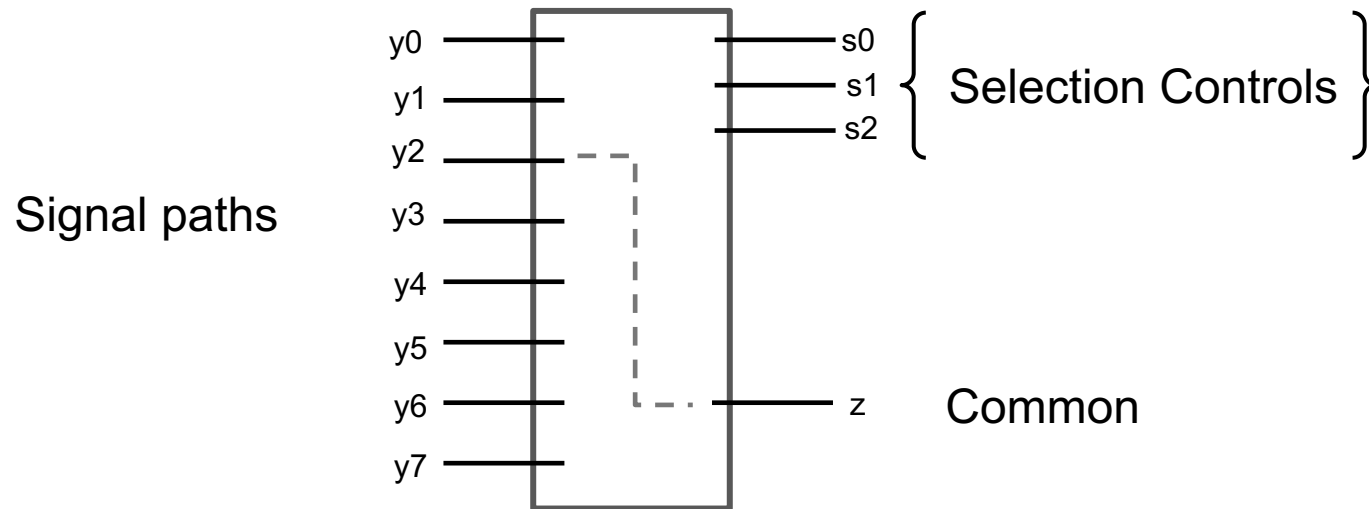
**An older family of ICs, which have a slightly different numbering scheme**

- **Manufacturer:** normally two letters.
- **Logic series:** we call this the “4000 series,” so this will always be 4.
- **Device Type:** What the chip actually *does*. A 4021 is shift register.
- **Output type:** “B” indicates “buffered” outputs. This is almost always the case now. It is a legacy from very early IC manufacturing. Don’t worry about it.
- **Other:** There can be several letters here, which give info about the package (surface mount, DIP, etc.), as well as the composition (lead free, “eco”), humidity/temperature ranges.

# WHAT IS A MULTIPLEXER?

**A multiplexer is essentially a digitally addressable “gate.”**

**Allows one of several signal paths to be connected to a common input/output.**



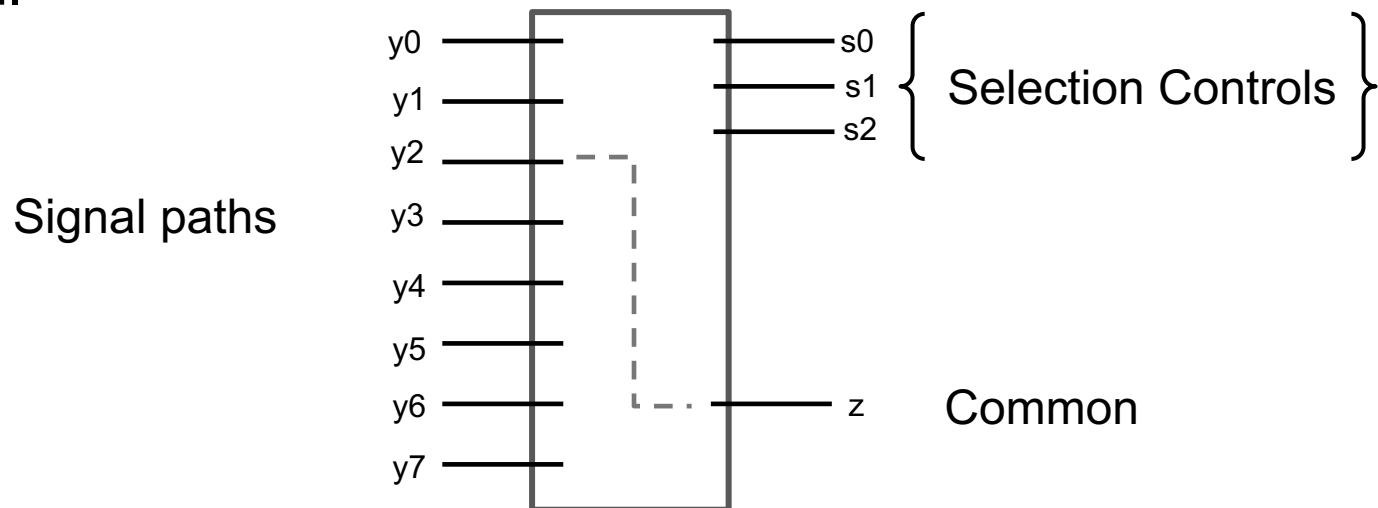
# MULTIPLEXING

The “Selection Control” pins allow you to set any one of the ***y<sub>n</sub>*** pins to the Common pin. In this example, it is configured so that pin “***y<sub>2</sub>***” is connected to the Common pin.

Different voltages on ***s<sub>0</sub>***, ***s<sub>1</sub>***, ***s<sub>2</sub>***, would connect a different ***y<sub>n</sub>*** to Common.

## FUN FACT

(Arduino’s Analog inputs are actually internally multiplexed. There is only one ADC in the Arduino. When we call `analogRead(n)`, the internal multiplexer routes pin *n* to the ADC.)



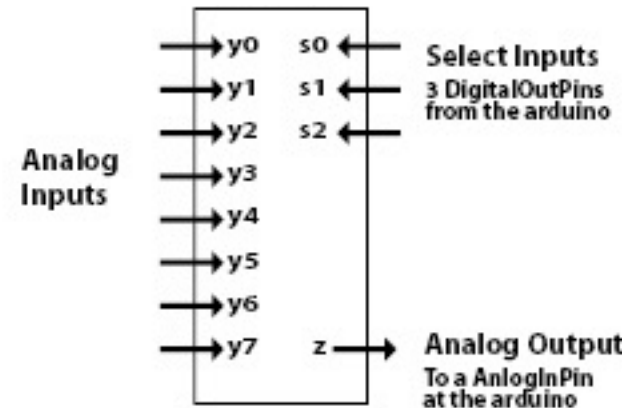
# MULTIPLEXING ANALOG INPUTS

A Mux is most commonly used to connect multiple analog inputs to a single ADC.

## Setup:

1. Connect 3 Digital output pins from Arduino to the “address pins” or “selection control” pins, labeled s0, s1, s2.
2. Connect the “Analog Output” or “Common” pin, labeled z, to an Analog Input pin on your Arduino.
3. Connect up to 8 analog voltages (e.g. from sensor circuits) to the the Analog Input pins, labeled y0, y1, y2, etc.
4. Any unused analog input pins should be connected to ground.

74HC4051 Multiplexer





# MULTIPLEXING ANALOG INPUTS

## Operation

1. Select which Input you want to measure by setting the 3 selection control pins HIGH or LOW with `digitalWrite()`. The values on the 3 selection control pins correspond to the input number, in binary. E.g., this would select pin `y3` (binary 011 is decimal number 3) :

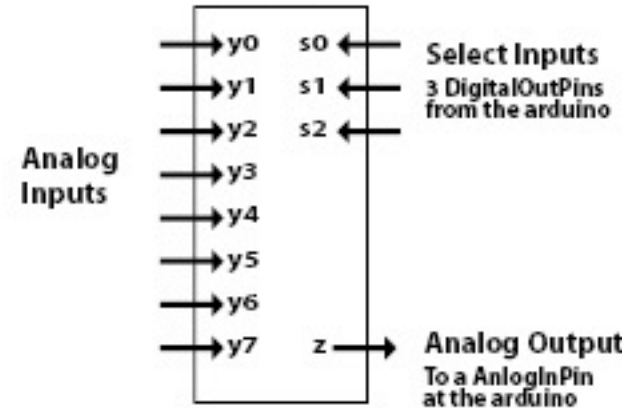
```
digitalWrite(s0,HIGH); // s0 is a digital pin set as an output
digitalWrite(s1,HIGH); // s1 is a digital pin set as an output
digitalWrite(s2,LOW);  // s2 is a digital pin set as an output
```

This will make an internal connection between the selected input pin (in this case `y3`) and the output pin (`z`).

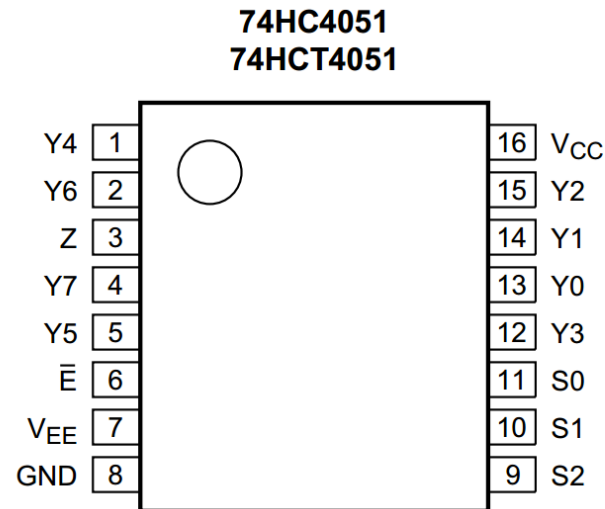
2. Take a reading on the Arduino analog input pin that is connected to pin `z`, i.e.:

```
analogRead(p); // p is the Arudino analog pin number
               //connected to pin z of the mux
```

74HC4051 Multiplexer



# 4051 PINOUT

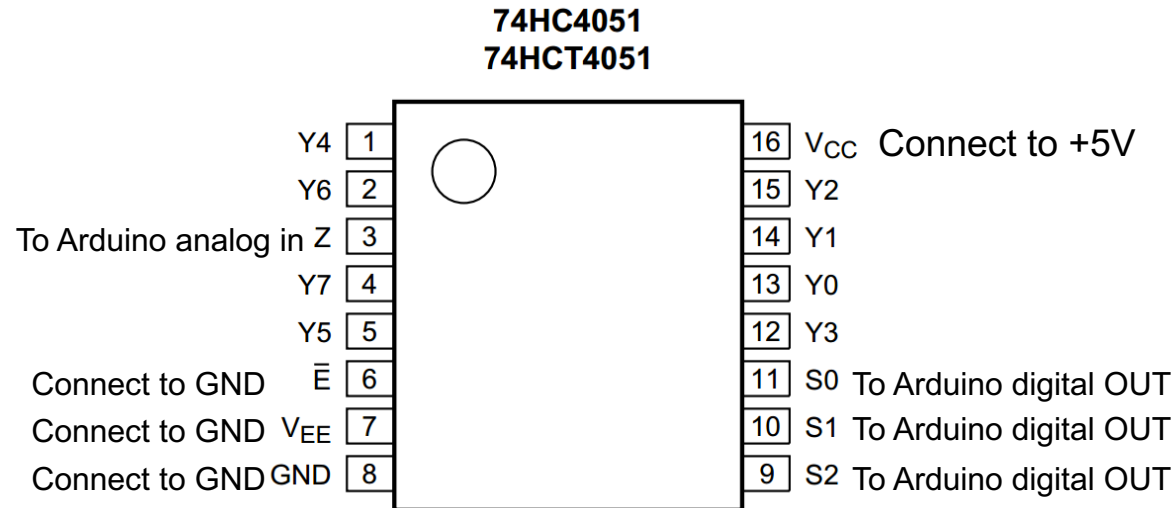


**Table 2. Pin description**

Symbol	Pin	Description
$\bar{E}$	6	enable input (active LOW)
V <sub>EE</sub>	7	supply voltage
GND	8	ground supply voltage
S0, S1, S2	11, 10, 9	select input
Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7	13, 14, 15, 12, 1, 5, 2, 4	independent input or output
Z	3	common output or input
V <sub>CC</sub>	16	supply voltage

# 4051 PINOUT

Inputs y0-y7 connect to analog sensors

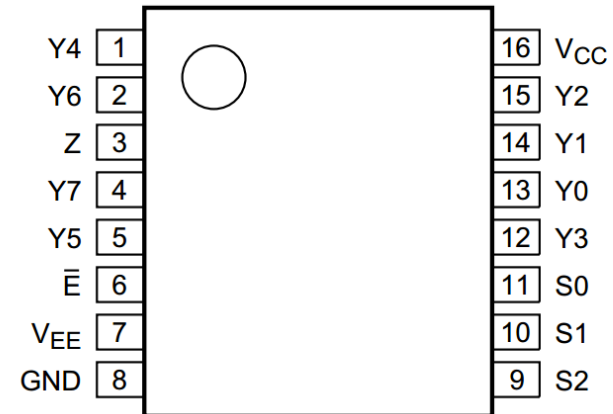


**Table 2. Pin description**

Symbol	Pin	Description
$\bar{E}$	6	enable input (active LOW)
V <sub>EE</sub>	7	supply voltage
GND	8	ground supply voltage
S0, S1, S2	11, 10, 9	select input
Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7	13, 14, 15, 12, 1, 5, 2, 4	independent input or output
Z	3	common output or input
V <sub>CC</sub>	16	supply voltage

# 4051 OPERATION

74HC4051  
74HCT4051



The  $\overline{E}$  is for “enable” and the bar over it means that its logic is inverted, i.e., it is “active” when “LOW” or at ground or 0V.

The  $\overline{E}$  would be read as “enable Low,” in other words, this pin needs to be Low for the chip to be enabled. Normally we just connect it to ground.

Table 3. Function table<sup>[1]</sup>

Input				Channel ON
$\overline{E}$	S2	S1	S0	
L	L	L	L	Y0 to Z
L	L	L	H	Y1 to Z
L	L	H	L	Y2 to Z
L	L	H	H	Y3 to Z
L	H	L	L	Y4 to Z
L	H	L	H	Y5 to Z
L	H	H	L	Y6 to Z
L	H	H	H	Y7 to Z
H	X	X	X	switches off

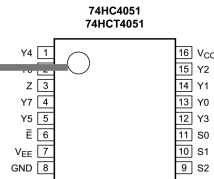
[1] H = HIGH voltage level;  
L = LOW voltage level;  
X = don't care.

# BASIC MUX EXAMPLE

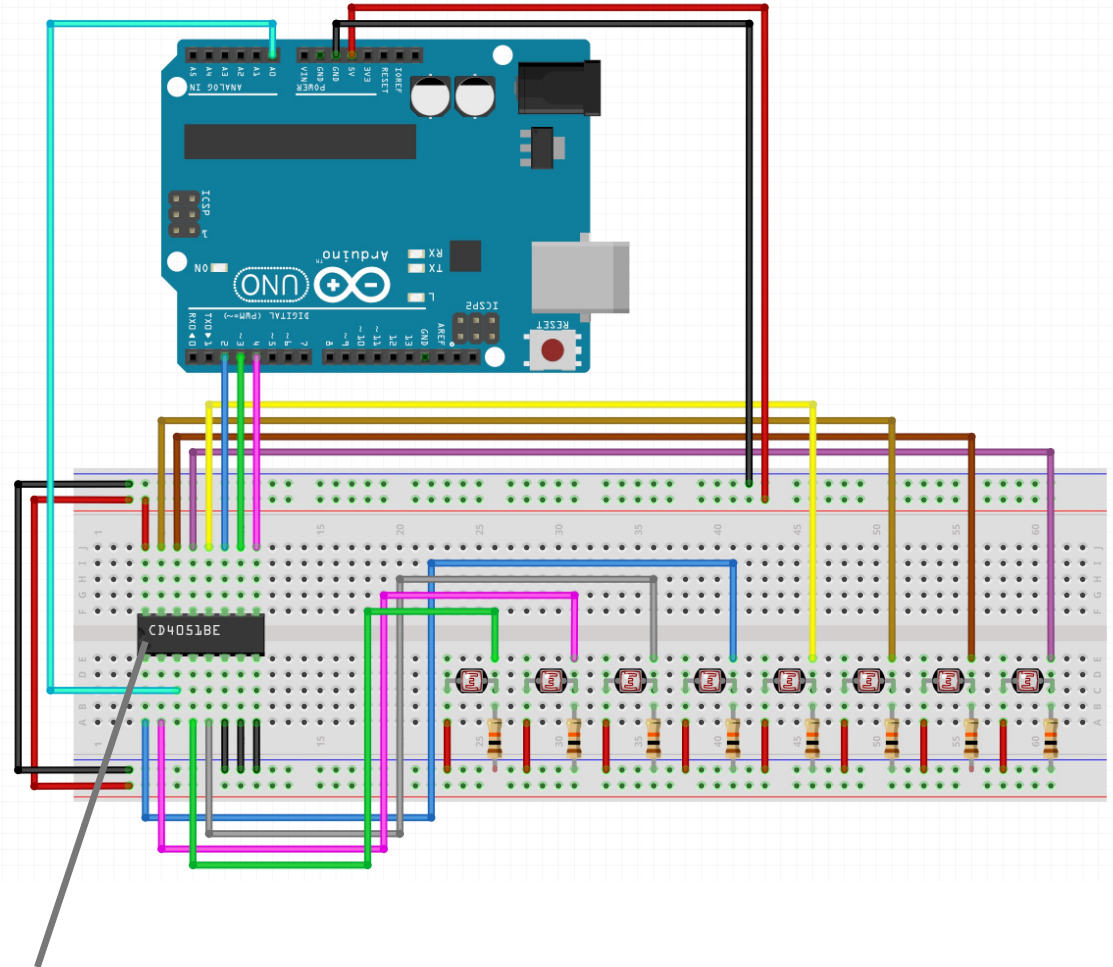
See mux.ino example.

This image shows graphically how you'd wire up the 4051, with 8 photocells as analog sensors.

“dot” indicates the top



There is always a “notch” on an IC in a DIP package, indicating the top of the chip, so you can identify pin 1.



“notch” on the chip indicates the top. Matches the “dot” on the pin diagram.

# MULTIPLE 4051S (ADVANCED)

The purpose of having an Enable pin is so that you can control multiple 4051 pins with the same 3 selection controls.

A single 4051 gives us 8 analog inputs for one physical input on the Arduino, and 3 digital outputs.

If we control the Enable pin with a digital output, we can then share the selection control pins between multiple 4051 chips. So for 2 4051s, we'd need 5 digital outputs: 3 for selection control and 2 for Enable controls. For 3 4051s, we'd need 6 digital outputs: 3 for selection control and 2 for Enable. We only add 1 digital output for each additional 4051.

In Arduino, you'd use the additional digital outputs to ENABLE one 4051 at a time, and DISABLE all others. Then cycle through and sample the 8 inputs of each 4051 chip, one at a time.

# MULTIPLE 4051S (ADVANCED)

# 4051s	# 4051 analog input chans	Arduino analog inputs used	Add'l Arduino Analog Inputs	TOTAL POSSIBLE ANALOG INPUTS	Selection control pins (dig Out)	Enable pins (dig out)	Total dig out pins
1	8	1	5	13	3	0	3
2	16	2	4	20	3	1	4
3	24	3	3	27	3	2	5
4	32	4	2	34	3	3	6
5	40	5	1	41	3	4	7
6	48	6	0	48	3	5	8

Bottom Line: with **8 digital output pins**, you can control **6** 4051 chips, which would give you a total of **48 analog inputs to your Arduino**.

# ADVANCED MUX CODE

See `multimux.ino` example