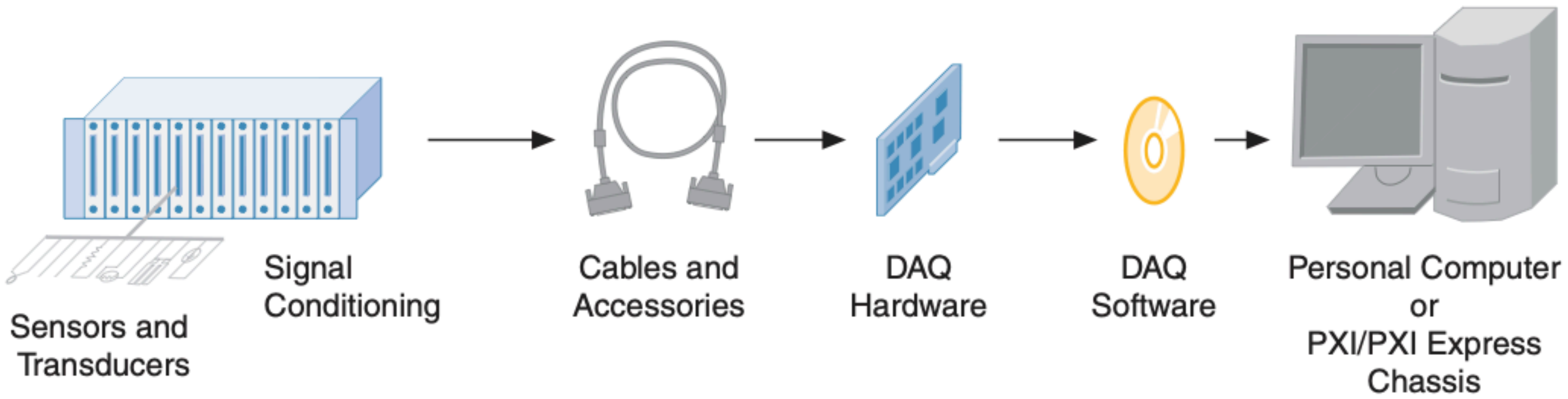
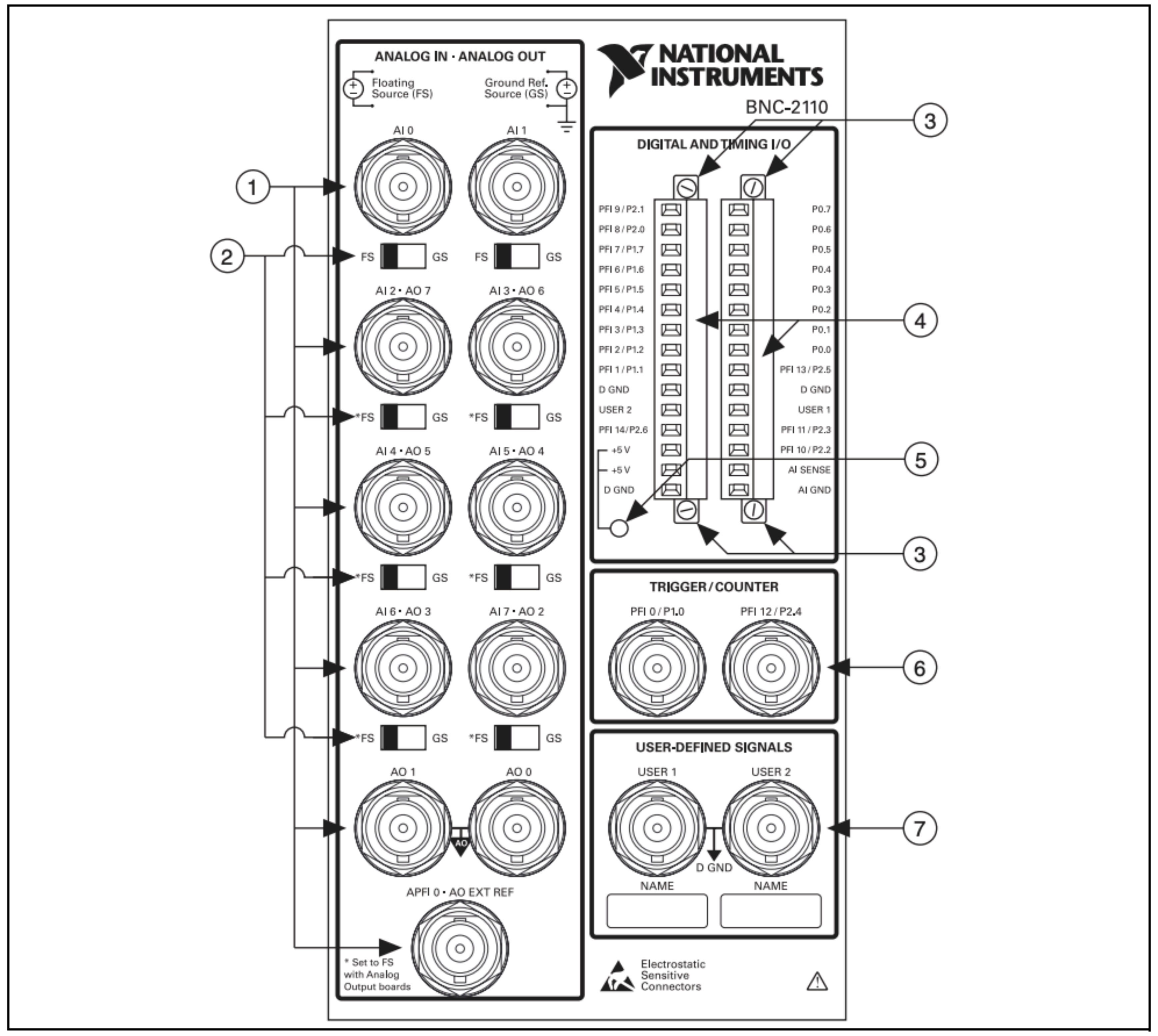
NI DAQ Overview

NI 6221 M Series Data Acquisition: 16-Bit, 250 kS/s, 16 AI, 24 DIO, 2 AO
NI PCIe-6321, 16 AI (16 bit, 250 kS/s), 2 AO (900 kS/s), 24 DIO

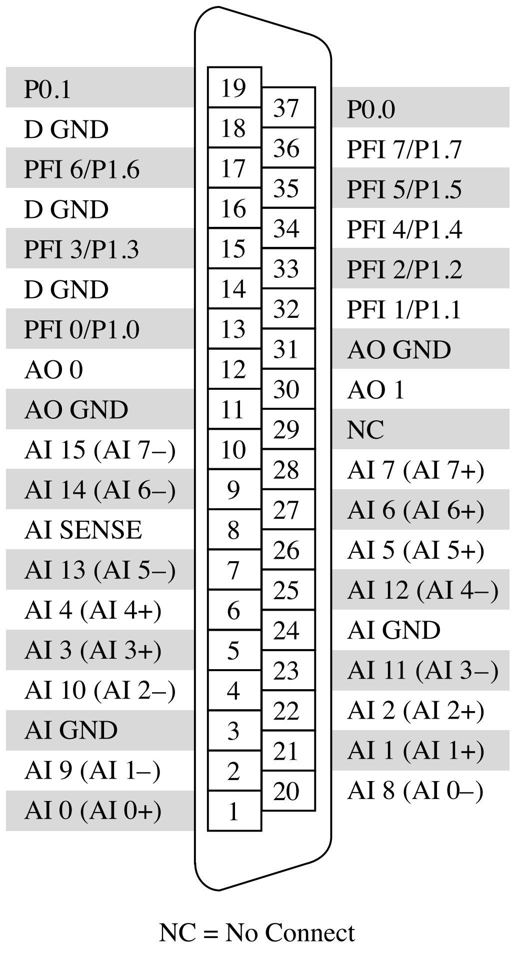
Is this nonsense?

Figure 2-1. Components of a Typical DAQ System





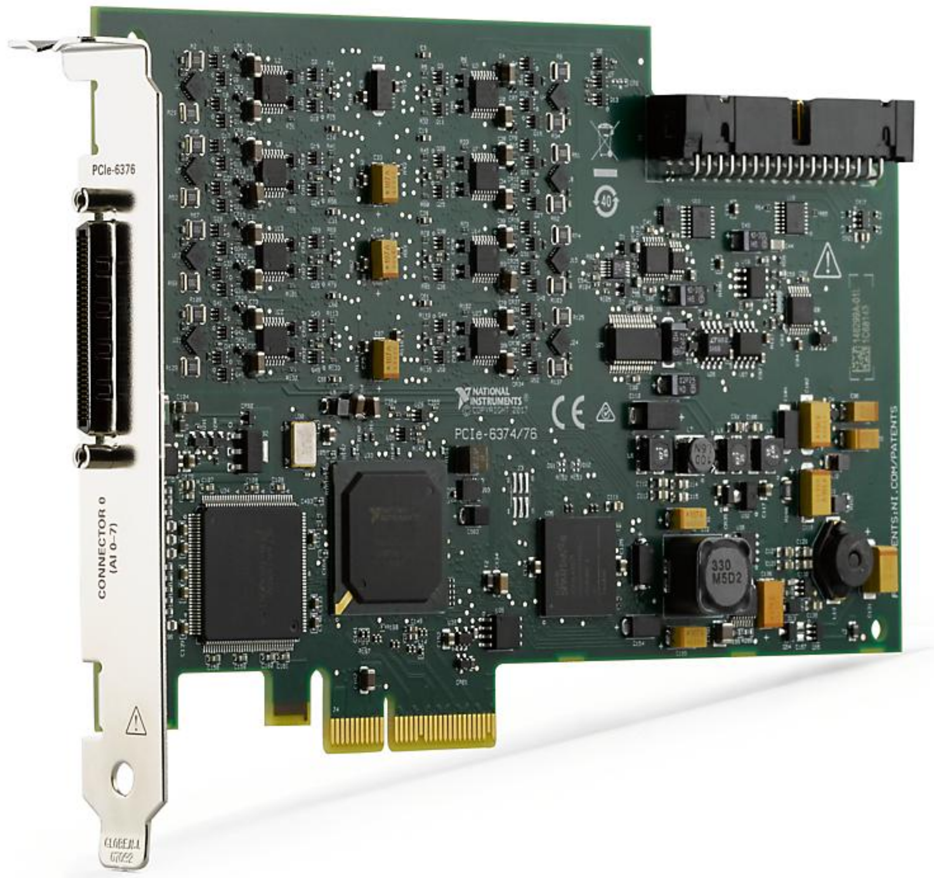
1	Analog Input/Analog Output BNC Connectors
2	FS/GS Switches
3	Terminal Block Retaining Screws
4	Digital and Timing I/O Spring Terminal Blocks
5	Power Indicator Light
6	Trigger/Counter BNC Connectors
7	User-Defined Signals BNC Connectors



Useful Links: PCIe-6321 Specs
NI-DAQmx Python Documentation
Bonsai - DAQmx Documentation

Acquisition Boards: What we have

NI USB-6221	NI PCIe-6332
The USB-6221 is designed to be plug-and-play	Breakout Board Required (BNC-2110)
Integrated Signal Conditioning	Scalability & Integration
Throughput and Latency	High Performance



NI DAQ Overview

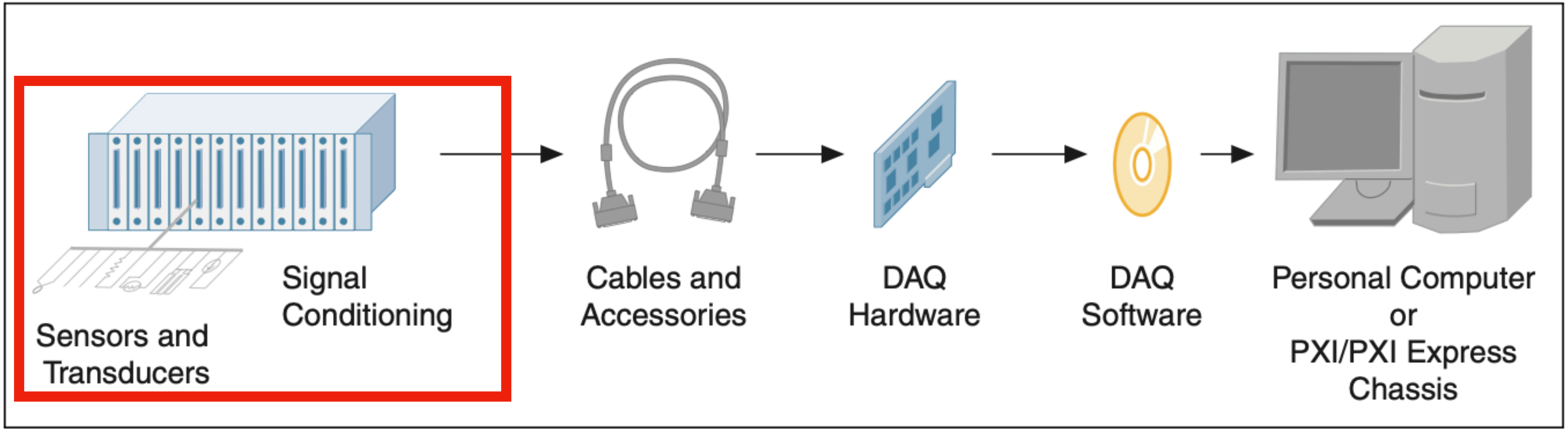


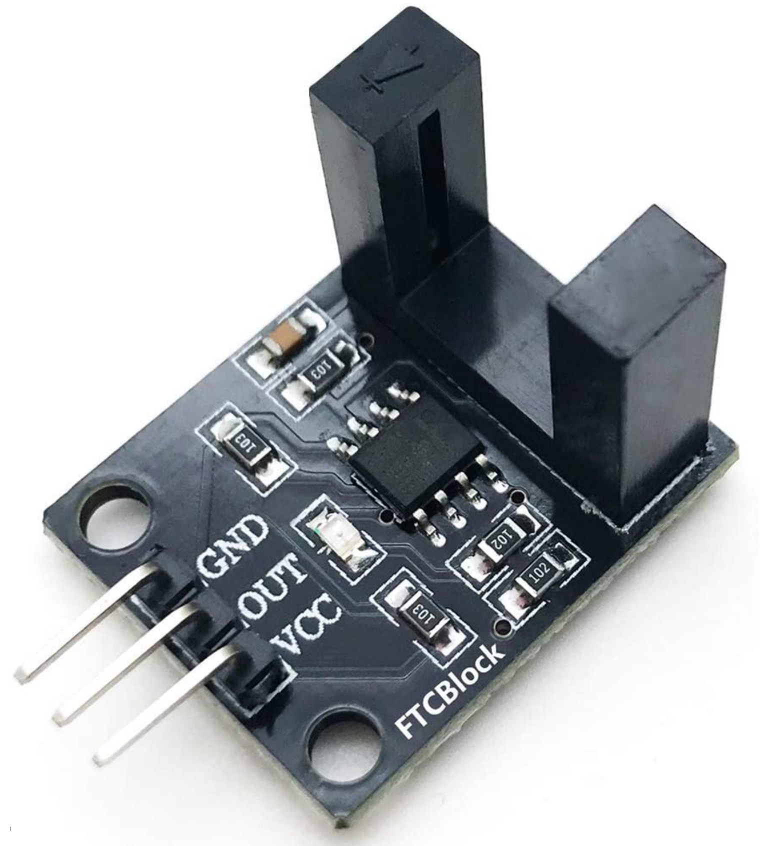
Figure 2-1. Components of a Typical DAQ System

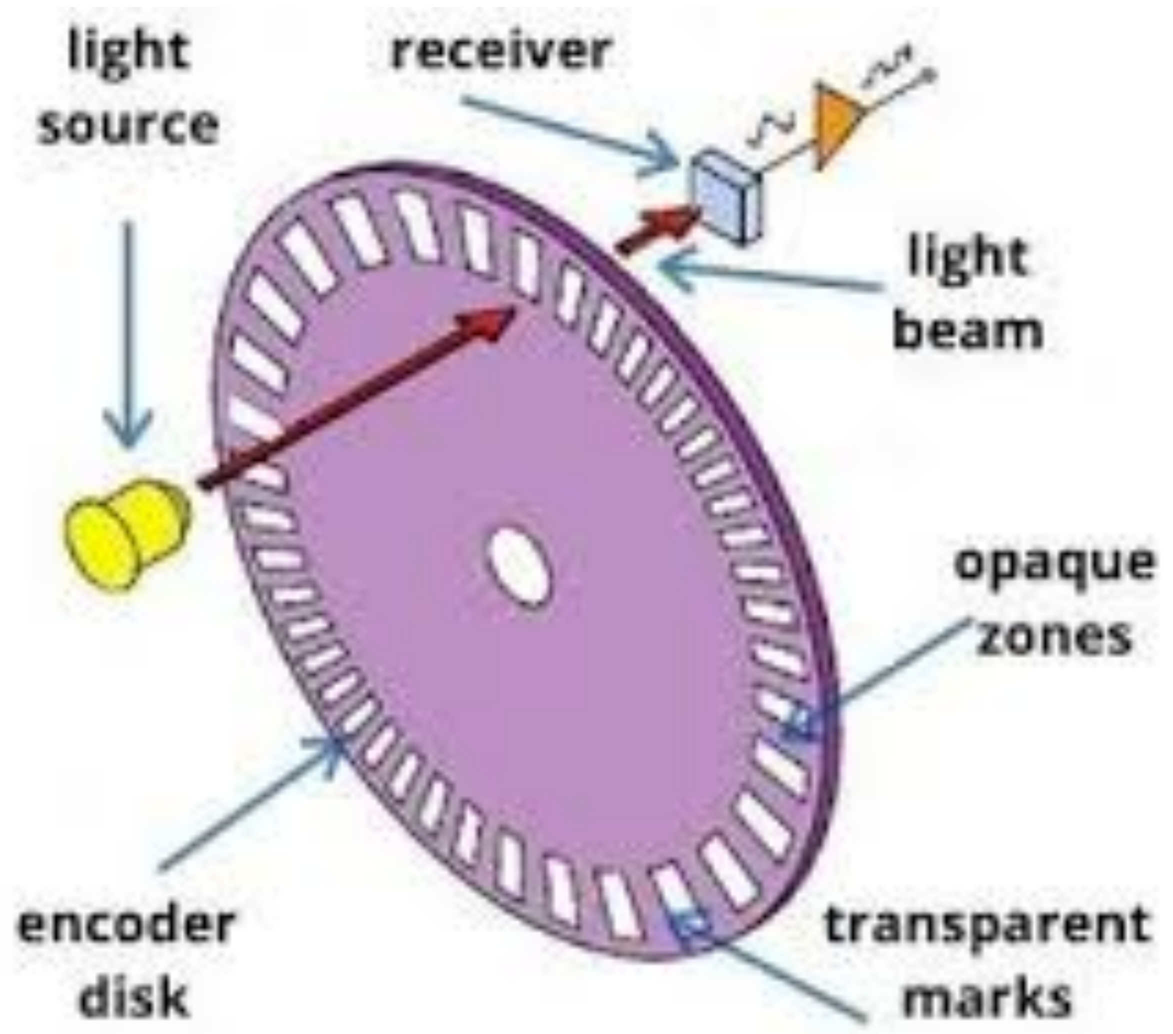
NI DAQ Overview

Sensor and Transducers

Sensors can generate electrical signals to measure physical phenomena, such as temperature, force, sound, or light. Some commonly used sensors are strain gauges, thermocouples,photodiodes, angular encoders, and thermistor.

To measure signals from these various transducers, you must convert them into a form that a DAQ device can accept.
This is called <u>Signal conditioning</u>



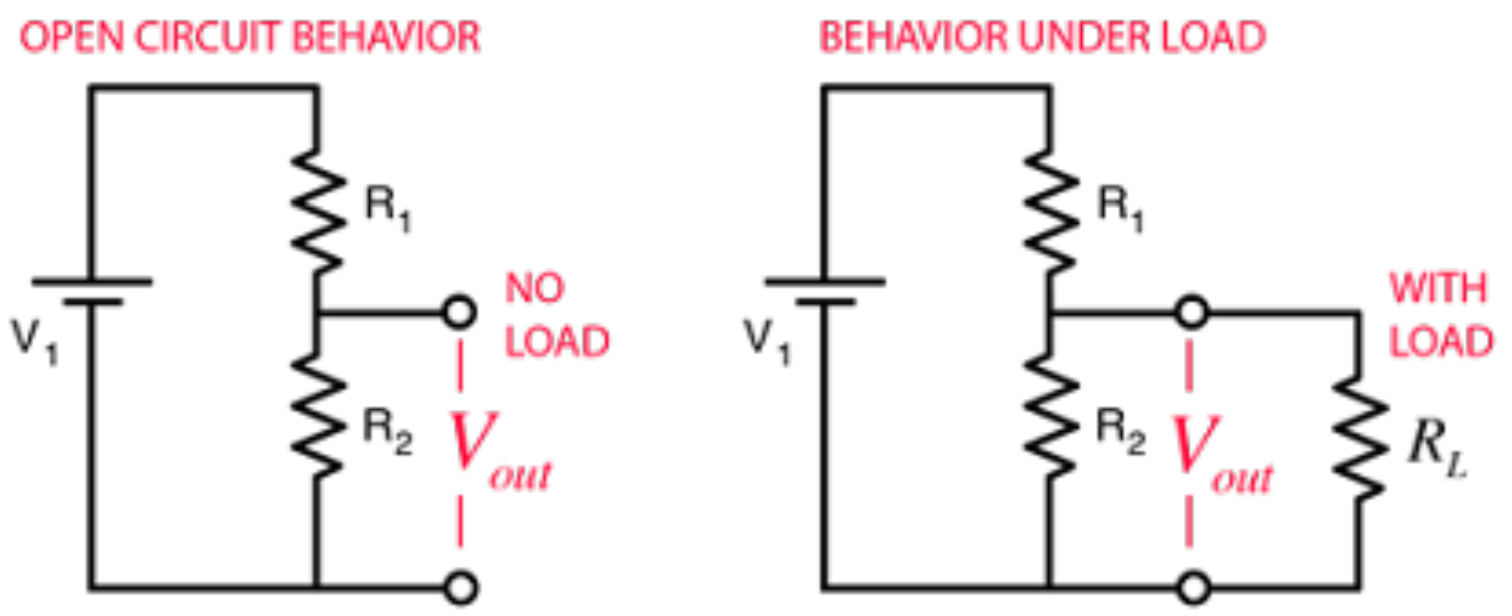


NI DAQ Overview

Signal conditioning

This mix of analog and digital signal conditioning components ensures that sensor outputs are amplified, filtered, and scaled to optimize the NI-DAQ's performance in precision measurement applications.

Common amplifiers architectures

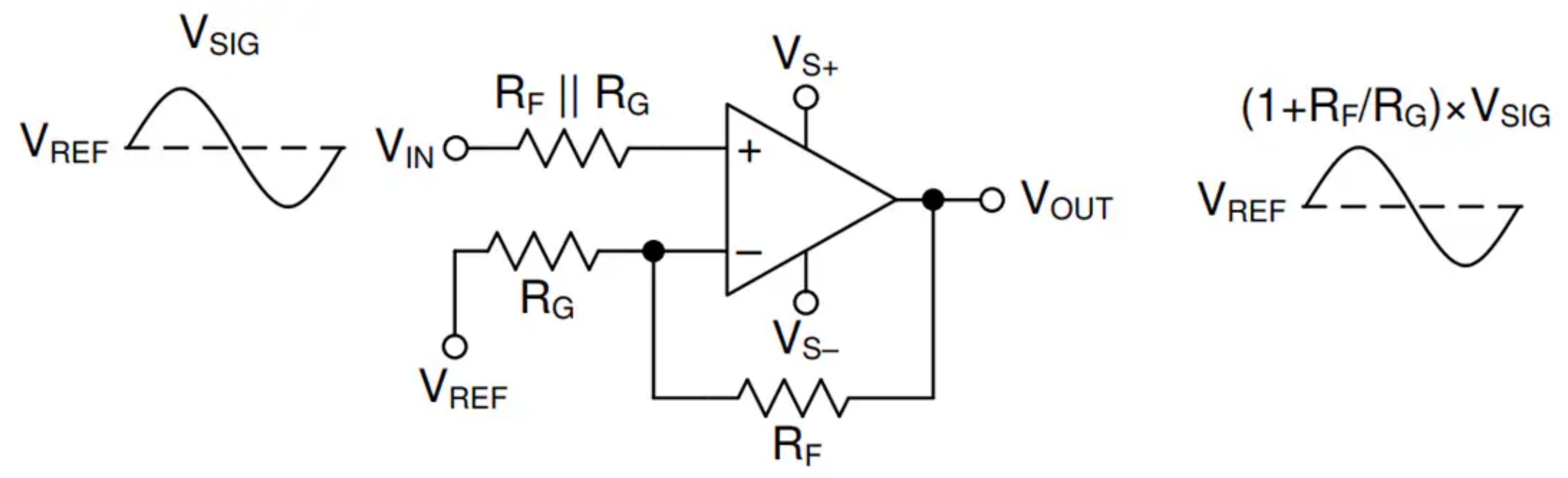


$$V_{out} = V_1 \frac{IR_2}{I(R_1 + R_2)} = \frac{V_1 R_2}{(R_1 + R_2)}$$

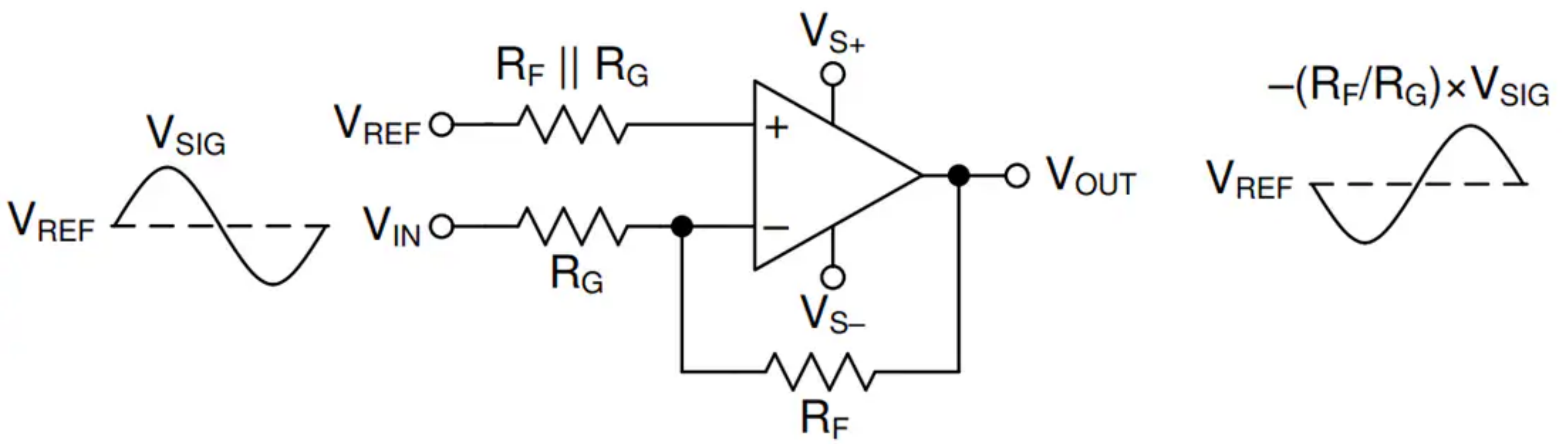
OUTPUT VOLTAGE UNDER "NO LOAD" CONDITION (open circuit)

OUTPUT VOLTAGE UNDER LOAD

$$V_{out} = V_1 \frac{IR_2}{I(R_1 + R_2)} = \frac{V_1 (R_2 \parallel R_L)}{(R_1 + R_2 \parallel R_L)}$$



Non-Inverting Amplifier



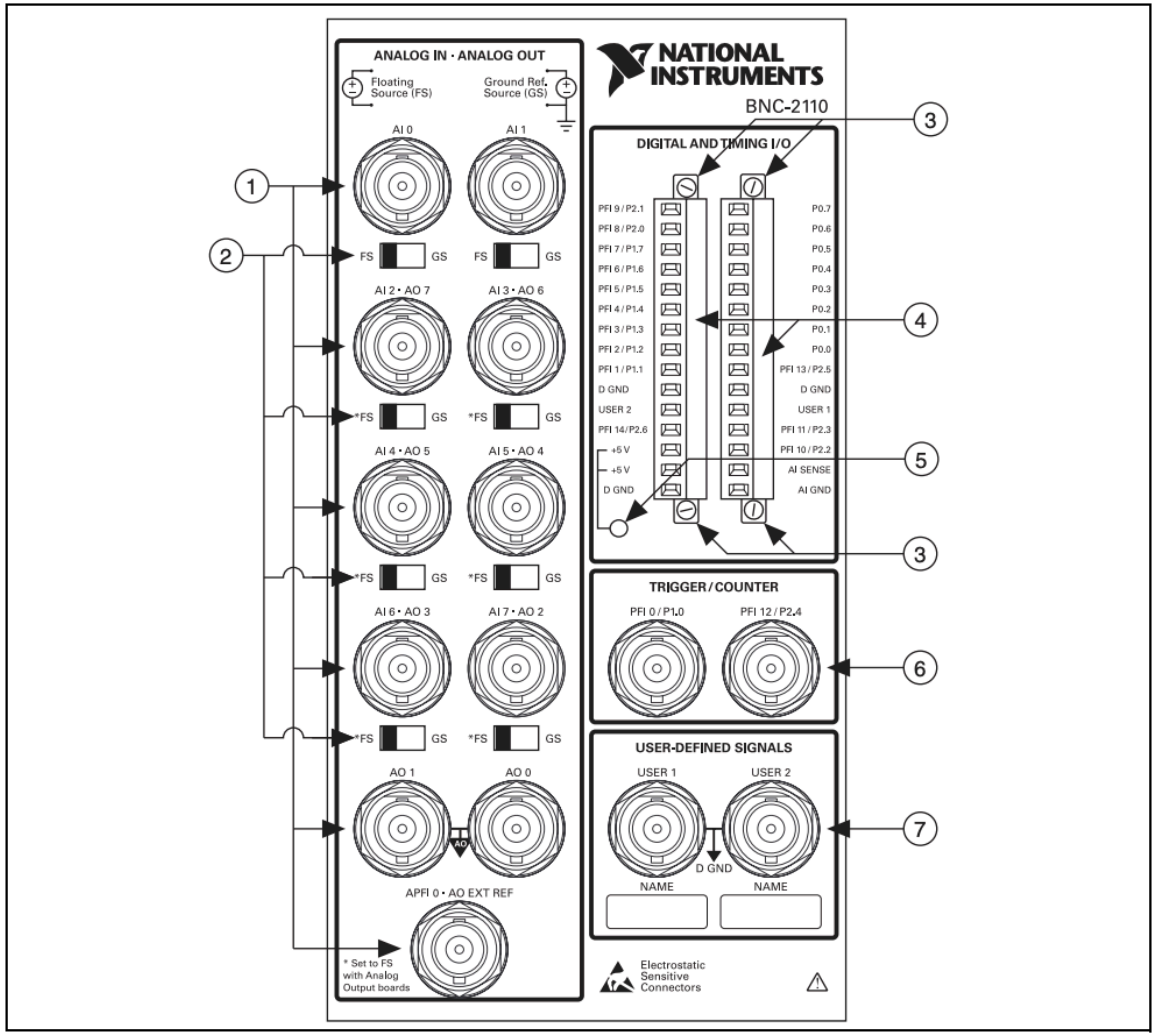
Inverting Amplifier

NI DAQ Overview

BNC-2110 Breakout board

Connector between your measurement apparatus and your DAQ device in laboratory

- 15 BNC connectors for analog input, analog output, trigger/counter functions, and user-defined signals
- A spring terminal block with 30 pins for digital and timing I/O signal connections



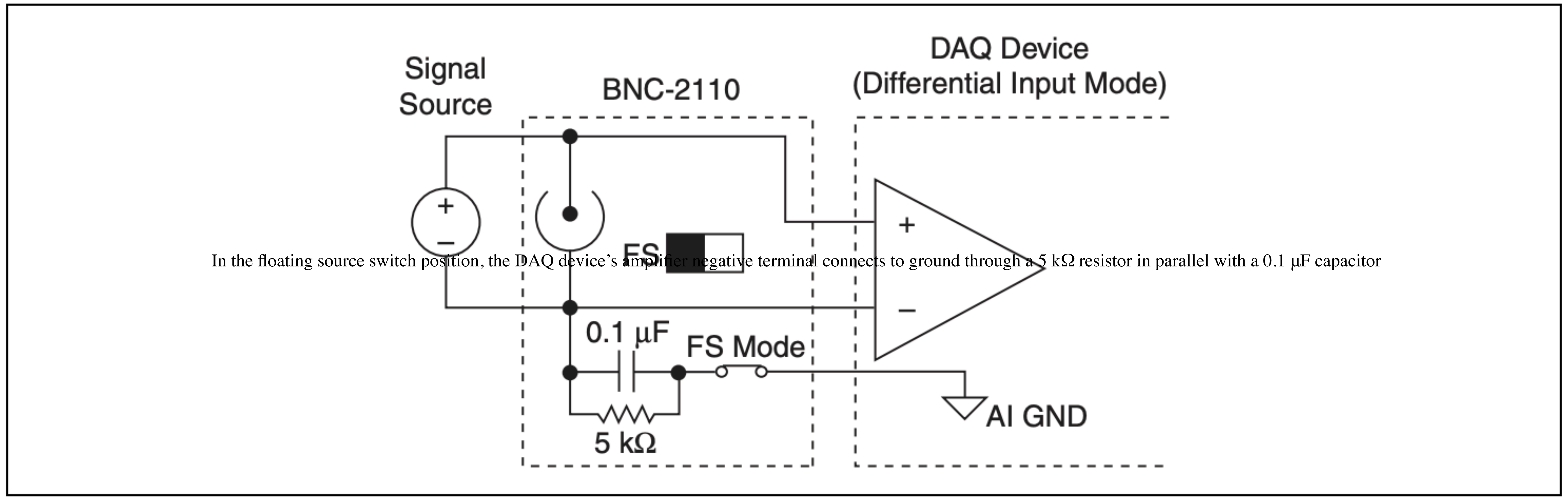
1	Analog Input/Analog Output BNC Connectors	5	Power Indicator Light
2	FS/GS Switches	6	Trigger/Counter BNC Connectors
3	Terminal Block Retaining Screws	7	User-Defined Signals BNC Connectors
4	Digital and Timing I/O Spring Terminal Blocks		

Figure 1. BNC-2110 Front Panel

NI DAQ Overview

Measuring floating signals

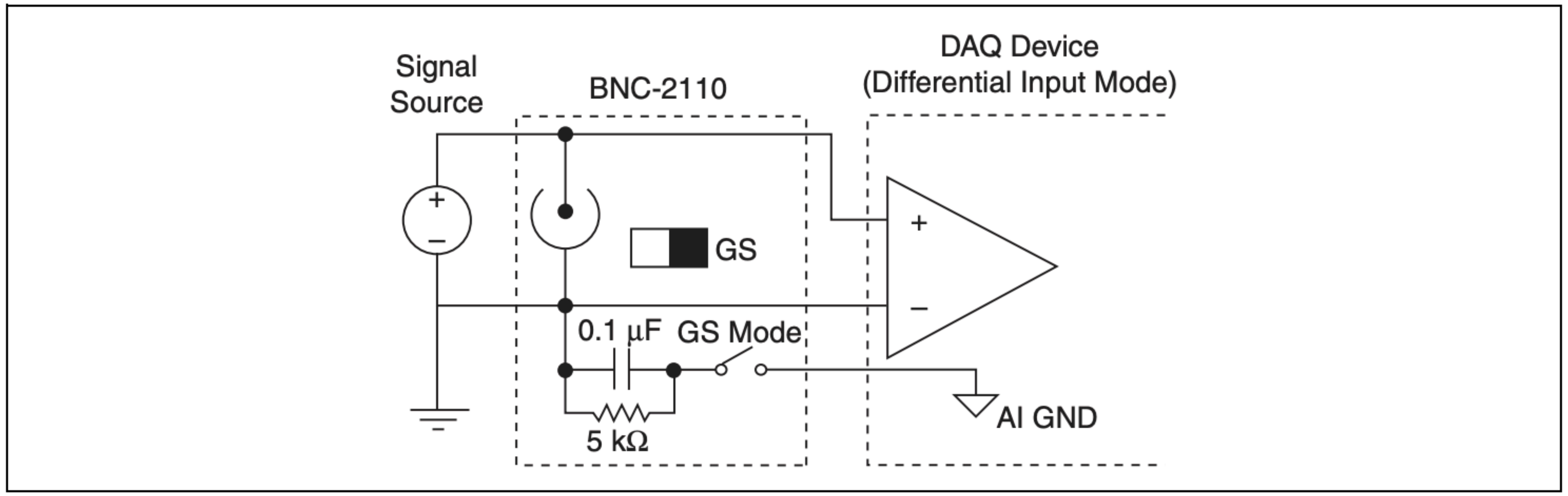
In the floating source switch position, the DAQ device’s amplifier negative terminal connects to ground through a 5 kΩ resistor in parallel with a 0.1 µF capacitor



In the floating source switch position, the DAQ device’s amplifier negative terminal connects to ground through a 5 kΩ resistor in parallel with a 0.1 µF capacitor

Measuring ground ref signals

In the ground reference mode the RC circuit is not effective and there is no bandwith filter



NI DAQ Overview

Analog output signals

When using connectors AO <2..7>, you must move the associated FS/GS switch(es) to the FS position.

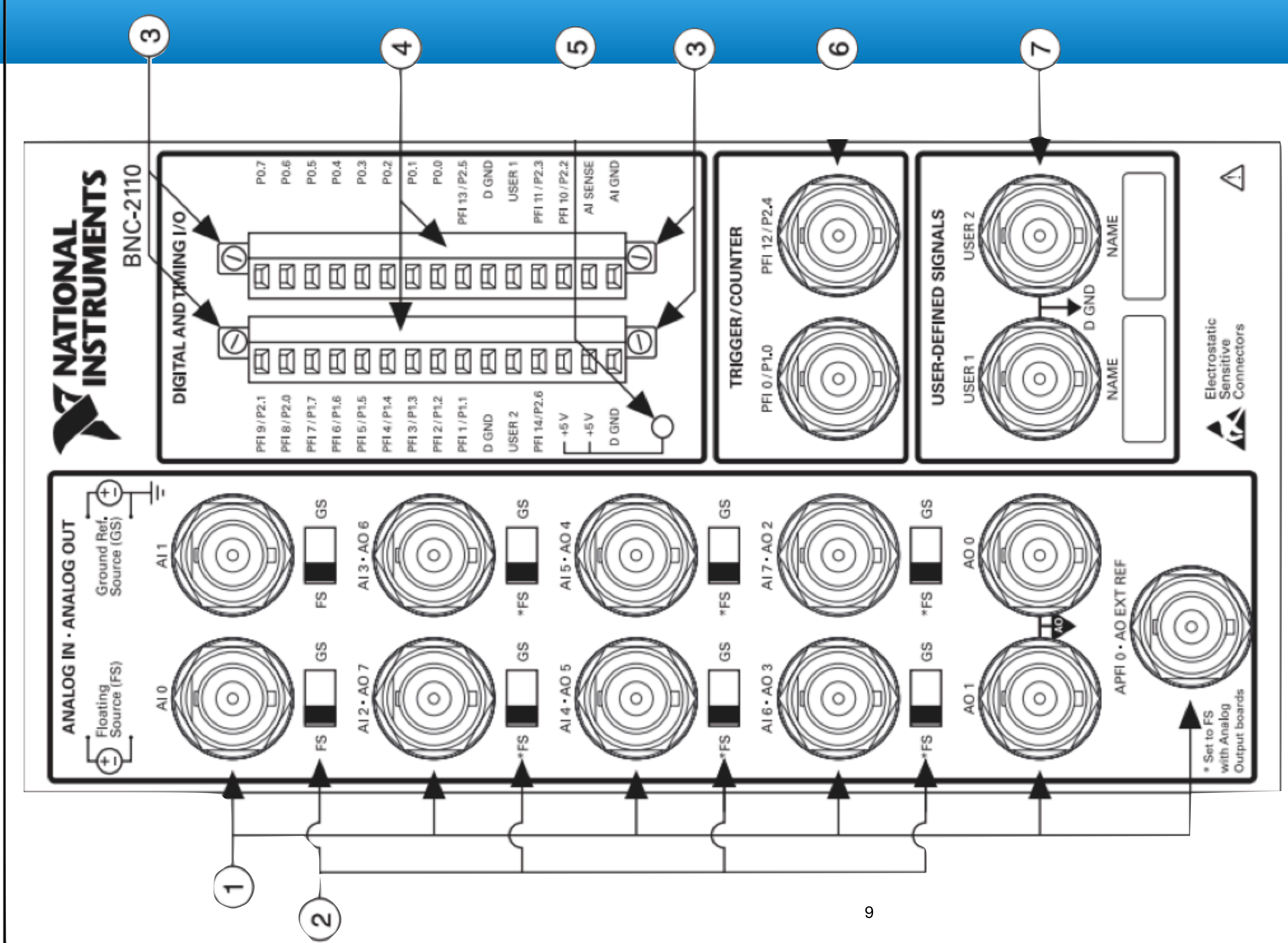
APFI 0/AO EXT REF signals

2 The AO EXT REF BNC is the external reference input for the AO circuitry. The APFI 0 channel can be used as the external reference input for the AO circuitry, the external offset for the AO circuitry, or the analog trigger input

Trigger/Counter Signals

- CTR 0 OUT (Counter 0 Output Signal)—As an input, this pin can be used to route signals directly to the RTSI bus. As an output, this pin emits the Ctr0InternalOutput signal. (Syncronization with other NI DAQ)

- **AI START TRIG** (AI Start Trigger Signal)—As an output, this pin is the ai/StartTrigger signal. In post-trigger DAQ sequences, a low-to-high transition indicates the initiation of the acquisition sequence.



1 Analog Input/Analog Output BNC Connectors

2 FS/GS Switches

3 Terminal Block Retaining Screws

4 Digital and Timing I/O Spring Terminal Blocks

5 Power Indicator Light

6 Trigger/Counter BNC Connectors

7 User-Defined Signals BNC Connectors

NI DAQ Overview

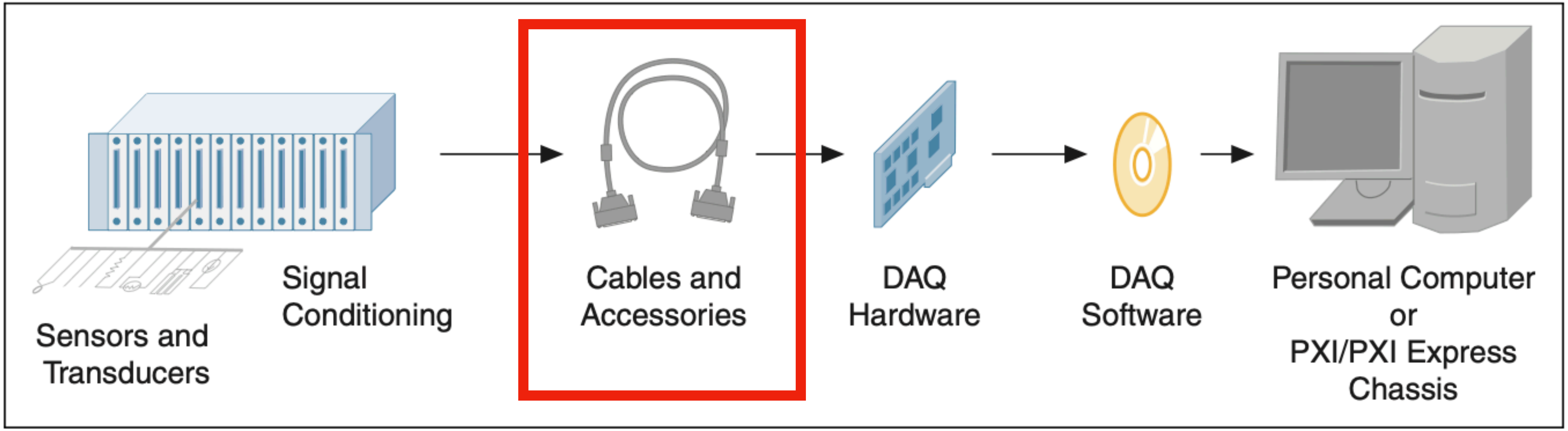
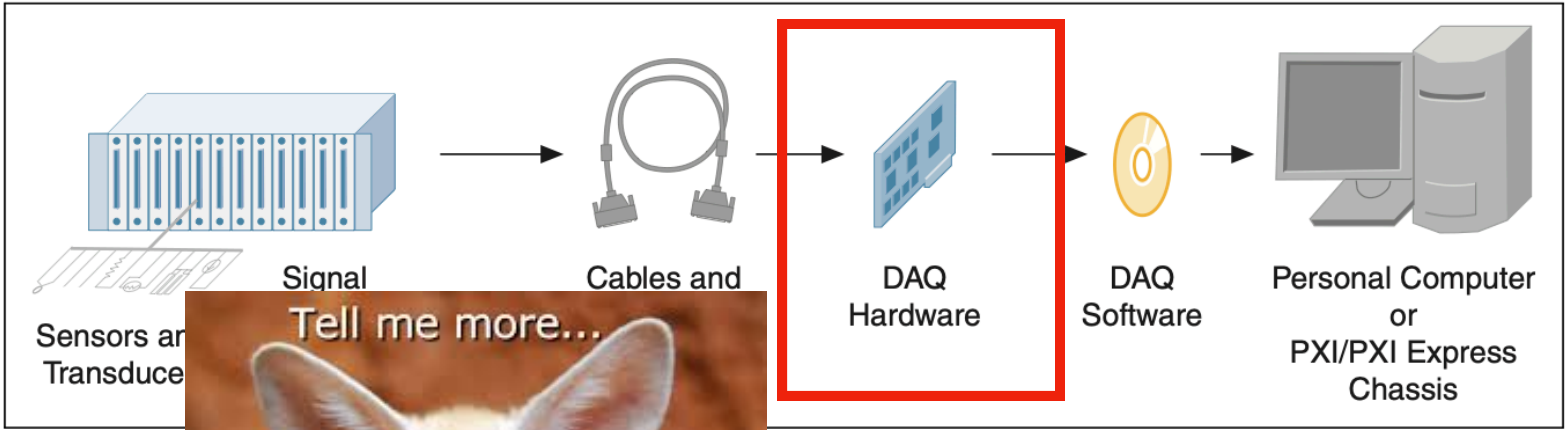


Figure 2-1. Components of a Typical DAQ System

Use the provided cable

NI DAQ Overview

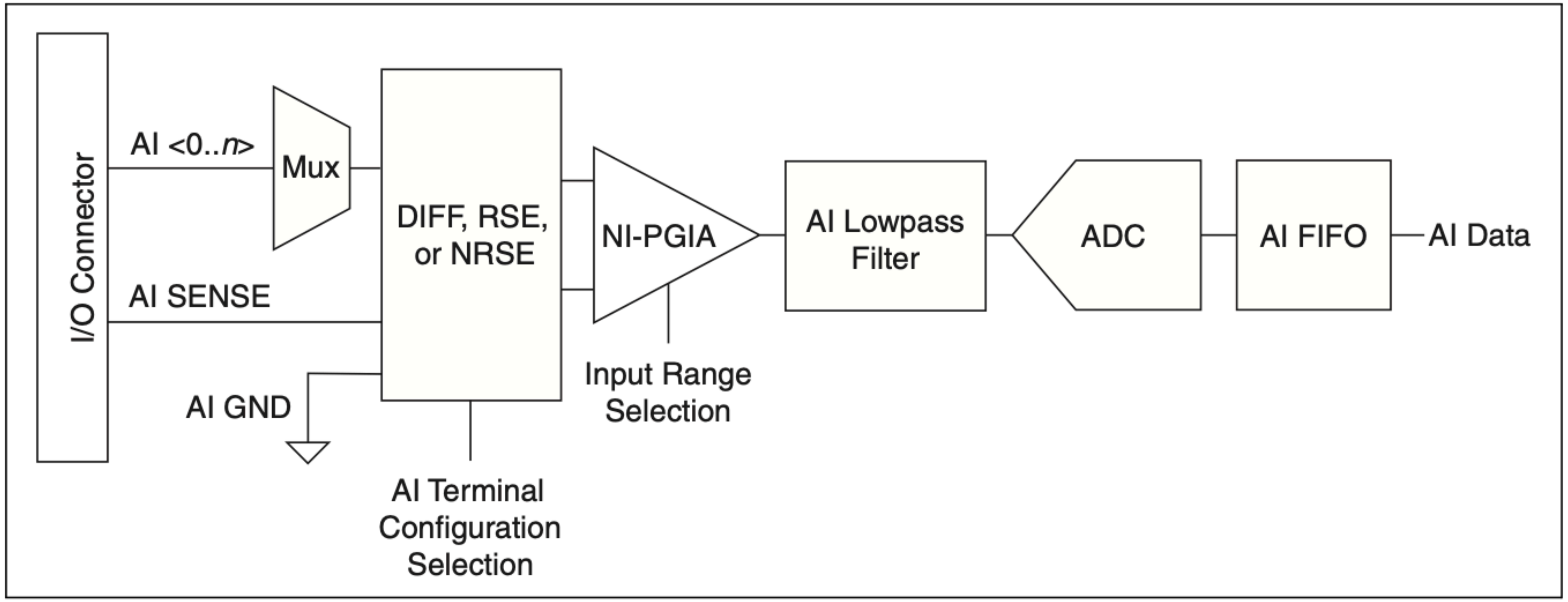


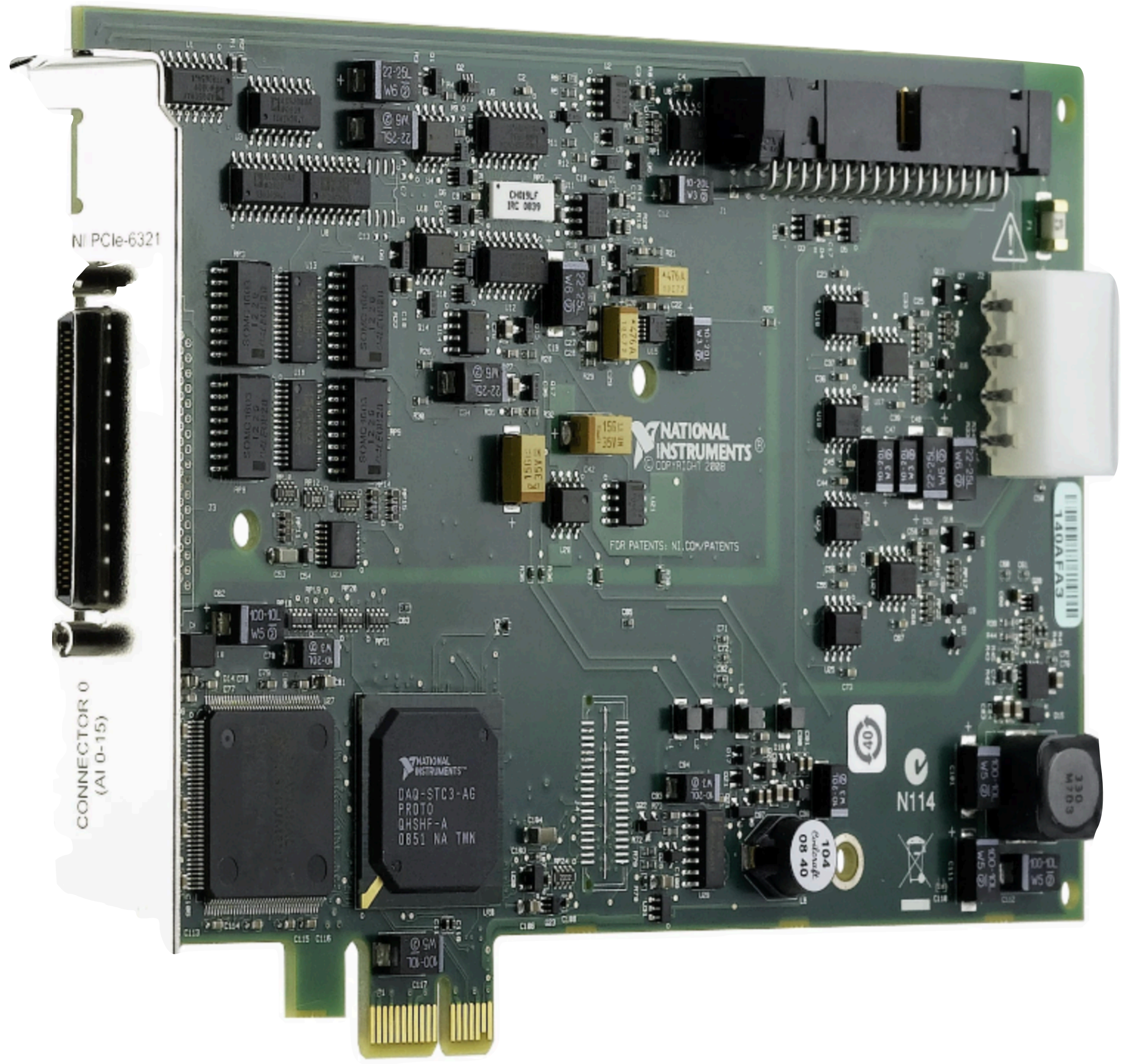
1. Components of a Typical DAQ System

NI DAQ Overview

PCIe-6321

DAQ hardware digitizes signals, performs D/A conversions to generate analog output signals, and measures and controls digital I/O signals.



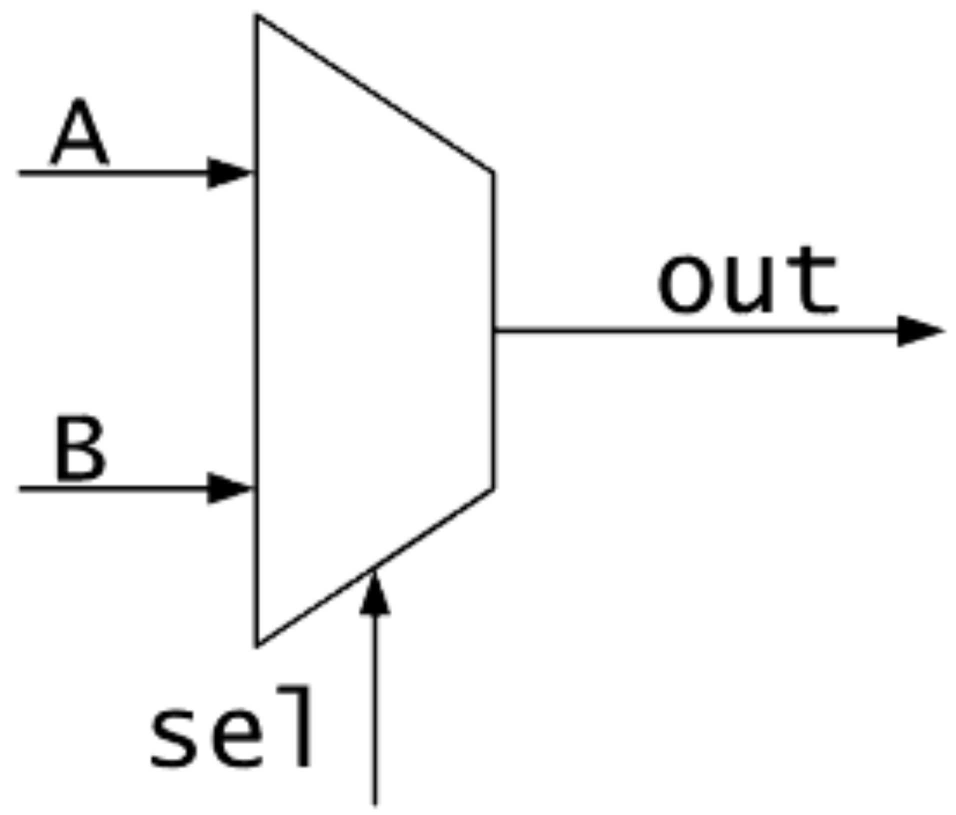


Each device has one analog-to-digital converter(ADC). The multiplexers (MUX) route one AI channel at a time tothe ADC through the NI-PGIA

MUX (Multiplexer)

A multiplexer (MUX) is a digital device that selects one input from multiple signals and forwards it to a single output line.

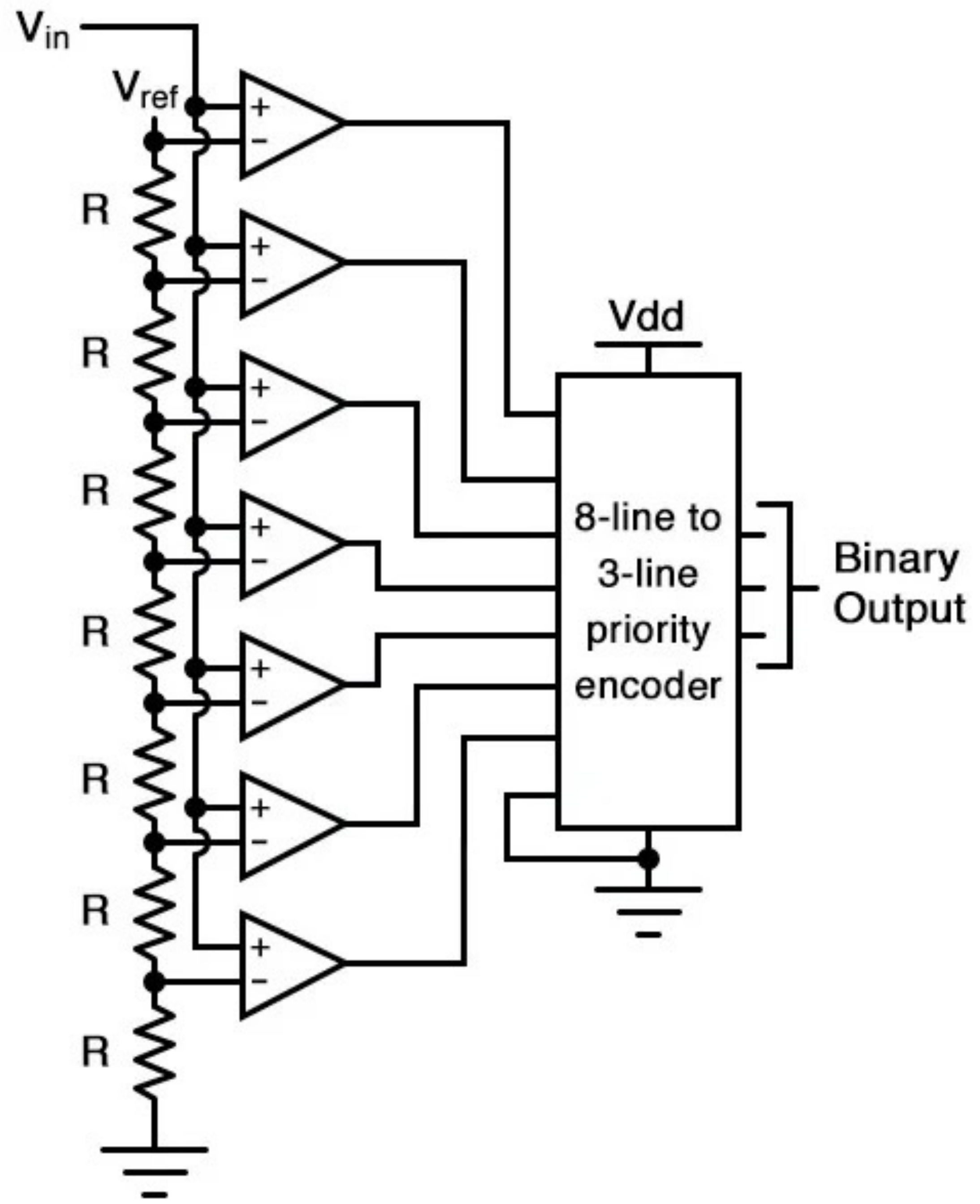
In a DAQ system, a multiplexer might allow one channel of the DAQ to sample from several sensors by rapidly switching between them based on control logic.



ADC (Analog to digital converter)

The 16-bit ADC converts analog inputs into one of 65,536 values. These values are spread fairly evenly across the input range. So, for an input range of –10 V to 10 V, the voltage of each code of a 16-bit ADC is:

$$\frac{(10\text{ V} - (-10\text{ V}))}{2^{16}} = 305\ \mu\text{V}$$



NI DAQ Overview

PCIe 6321 Analog Input Specs

- **Channels:** 16 analog input channes, 16 single ended or 8 differential
- **Resolution:** 16-bit
- **Sampling Rate:** 250 kSample/s Single channel Maximum
- **Input Range:** ±0.2 V, ±1 V, ±5 V, ±10 V (programmable)
- **Additional Features:** Built-in anti-aliasing filtering and programmable gain options.

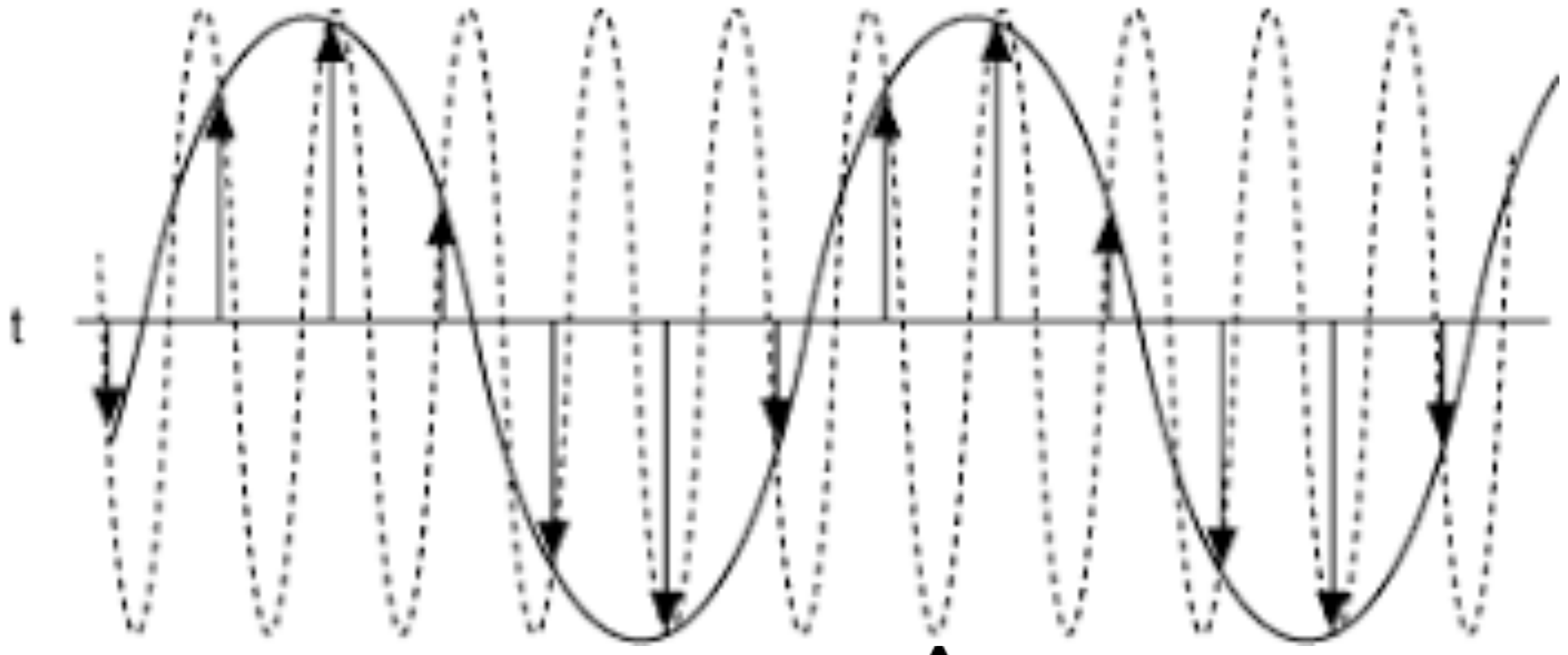
PCIe 6321 Analog Output Specs

- **Channels:** 2
- **Resolution:** 16-bit
- **Sampling Rate:** 1 channel 900 kSample/s 2 channels 840 kSample/s per channel
- **Input Range:** ±10 V

Sampling & Shannon's Theorem in NI-DAQ Systems

To accurately reconstruct a signal, the sampling rate must be at least twice the highest frequency component of the signal:

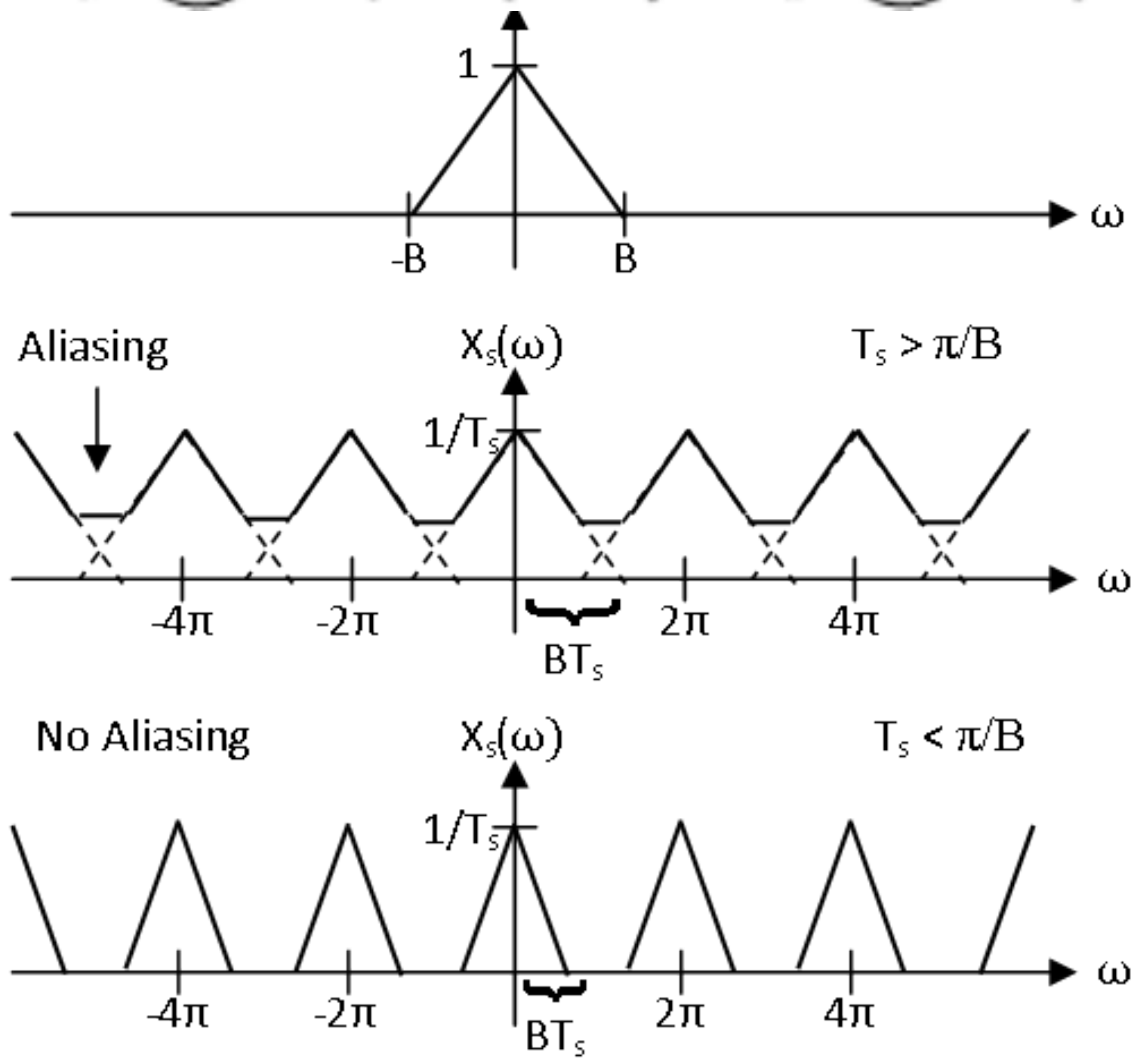
$$f_s \geq 2f_{\text{max}}$$



If this condition is not met, aliasing occurs, distorting the measured signal.

In NI-DAQ systems with multiplexed analog inputs, a single ADC is shared across multiple channels. The effective sampling rate per channel decreases as the number of active channels increases:

$$f_{\text{effective}} = \frac{f_s}{N}$$ where N is the number of channels used



NI DAQ Overview

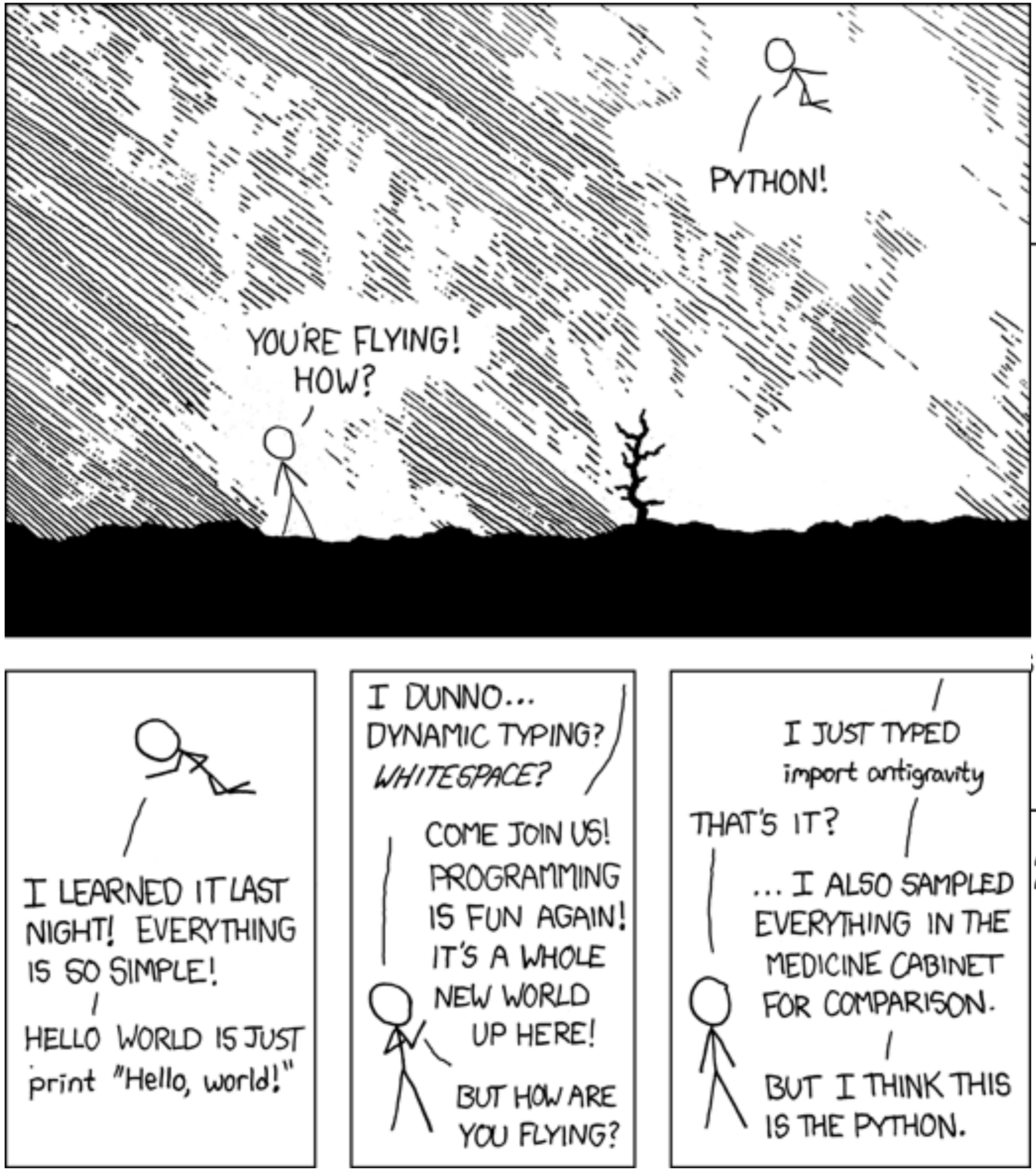
Digital I/O & Timing (See BNC-2110)

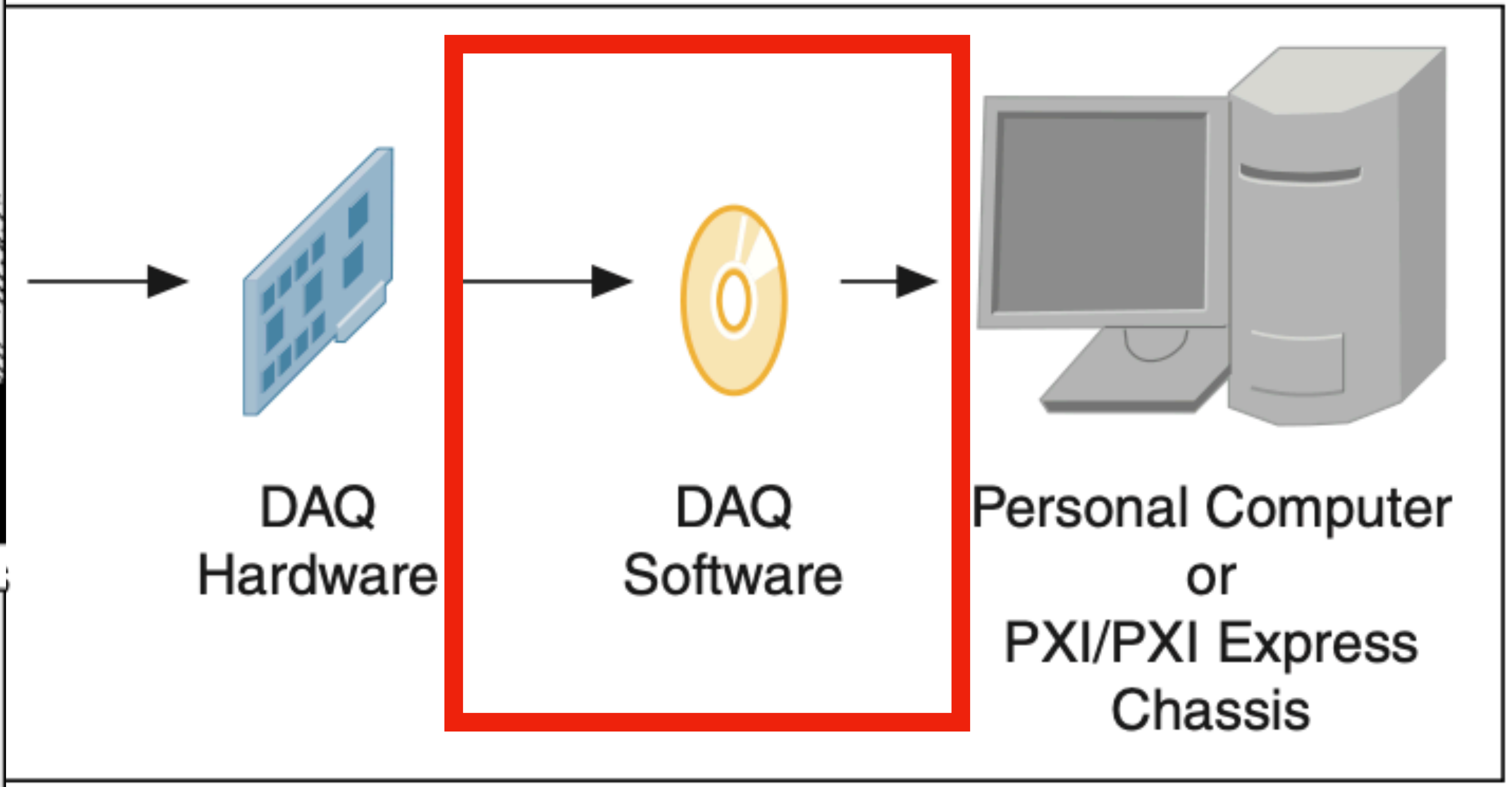
- Digital I/O Channels: Multiple digital channels for control and interfacing.
- Timing Capabilities: Counter, triggering inputs/outputs.
- Synchronization: Ability to route internal signals via RTSI for multi-device setups.

Triggering & Synchronization

- Analog & Digital Triggers: Support for flexible trigger sources (external trigger, analog start trigger).
- Programmable Timing: Configure timing parameters via NI-DAQmx.
- RTSI Bus Compatibility: Facilitates synchronization across multiple NI devices.

NI DAQ Overview





2-1. Components of a Typical DAQ System

NI DAQ Overview

```
import nidaqmx
from nidaqmx.constants import TerminalConfiguration, AcquisitionType

# Create a Task for analog input acquisition
with nidaqmx.Task() as task:
    # Add an analog input channel:
    # - "Dev1/ai0": The physical channel name
    # - Voltage range set to -10V to +10V to match your sensor's output and desired gain
    # - Differential configuration for improved noise rejection
    ai_channel = task.ai_channels.add_ai_voltage_chan(
        "Dev1/ai0", min_val=-10.0, max_val=10.0,
        terminal_config=TerminalConfiguration.DIFFERENTIAL
    )

    # Some devices support programmable gain settings.
    # If available, you could adjust the gain property here.
    # Example (if supported): ai_channel.ai_gain = 10.0

    print("Configured AI channel with range: {} to {} V".format(ai_channel.min_val,
ai_channel.max_val))

    # Configure timing: continuous acquisition at 10 kS/s, 1000 samples per channel
    task.timing.cfg_samp_clk_timing(rate=10000,
                                    sample_mode=AcquisitionType.CONTINUOUS,
                                    samps_per_chan=1000)

    # Read a block of data
    data = task.read(number_of_samples_per_channel=1000)
    print("Acquired Data:", data)
```

NI DAQ Overview

```python
import nidaqmx
from nidaqmx.constants import TerminalConfiguration, AcquisitionType, Edge

# Create a DAQmx Task (the container for channels and settings)
with nidaqmx.Task() as task:
    # 1. Add an Analog Input Channel:
    #    - "Dev1/ai0" is the physical channel.
    #    - Configured for differential measurement (good for rejecting noise).
    #    - The voltage range is set to -5V to +5V (narrower range gives higher effective gain).
    ai_channel = task.ai_channels.add_ai_voltage_chan(
        "Dev1/ai0", min_val=-5.0, max_val=5.0,
        terminal_config=TerminalConfiguration.DIFFERENTIAL
    )

    # 2. Configure the Sample Clock:
    #    - Sets the sampling rate to 20 kS/s.
    #    - Acquires 1000 samples per channel per read.
    #    - Continuous mode means the acquisition continues until stopped.
    task.timing.cfg_samp_clk_timing(
        rate=20000,
        sample_mode=AcquisitionType.CONTINUOUS,
        samps_per_chan=1000
    )

    # 3. Set Up an Analog Edge Start Trigger:
    #    - Trigger Source: "Dev1/ai1" (another channel acting as trigger).
    #    - Trigger Level: 1.0 V (acquisition starts when this threshold is exceeded).
    #    - Trigger Slope: Rising edge.
    #    This configuration waits for the voltage on channel ai1 to exceed 1V on the rising edge.
    task.triggers.start_trigger.cfg_anlg_edge_start_trig(
        trigger_source="Dev1/ai1",
        trigger_level=1.0,
        trigger_slope=Edge.RISING
    )

    print("Waiting for trigger event on Dev1/ai1...")

    # 4. Read a Block of Data:
    #    - Once triggered, read 1000 samples from the analog input channel.
    data = task.read(number_of_samples_per_channel=1000)
    print("Acquired Data:", data)
```