Importing some libraries

In [1]:

```
from pynq.overlays.base import BaseOverlay
import pynq.lib.rgbled as rgbled
import time
```

Programming the PL

In [2]:

```
base = BaseOverlay("base.bit")
```

Defining buttons and LEDs

In [3]:

```
btns = base.btns_gpio
led4 = rgbled.RGBLED(4)
led5 = rgbled.RGBLED(5)
```

Using a loop to blink the LEDS and read from buttons

In [15]:
```
#Press a button to turn off the LED flashing!
while True:
    led4.write(0x1)
    led5.write(0x7)
    if btns.read() != 0:
        break
    time.sleep(0.1)
    led4.write(0x0)
    led5.write(0x0)
    if btns.read() != 0:
        break
    time.sleep(0.05)
    led4.write(0x1)
    led5.write(0x7)
    if btns.read() != 0:
        break
    time.sleep(0.1)
    led4.write(0x0)
    led5.write(0x0)
    if btns.read() != 0:
        break
    time.sleep(0.05)

    led4.write(0x7)
    led5.write(0x4)
    if btns.read() != 0:
        break
    time.sleep(0.1)
    led4.write(0x0)
    led5.write(0x0)
    if btns.read() != 0:
        break
    time.sleep(0.05)
    led4.write(0x7)
    led5.write(0x4)
    if btns.read() != 0:
        break
    time.sleep(0.1)
    led4.write(0x0)
    led5.write(0x0)
    if btns.read() != 0:
        break
    time.sleep(0.05)

led4.write(0x0)
led5.write(0x0)
```

Using asyncio to blink the LEDS and read from buttons

In [ ]:

```
import asyncio
cond = True

async def flash_leds():
    global cond, start
    while cond:
        led4.write(0x1)
        led5.write(0x7)
        await asyncio.sleep(0.1)
        led4.write(0x0)
        led5.write(0x0)
        await asyncio.sleep(0.05)
        led4.write(0x1)
        led5.write(0x7)
        await asyncio.sleep(0.1)
        led4.write(0x0)
        led5.write(0x0)
        await asyncio.sleep(0.05)

        led4.write(0x7)
        led5.write(0x4)
        await asyncio.sleep(0.1)
        led4.write(0x0)
        led5.write(0x0)
        await asyncio.sleep(0.05)
        led4.write(0x7)
        led5.write(0x4)
        await asyncio.sleep(0.1)
        led4.write(0x0)
        led5.write(0x0)
        await asyncio.sleep(0.05)

async def get_btns(_loop):
    global cond, start
    while cond:
        await asyncio.sleep(0.01)
        if btns.read() != 0:
            _loop.stop()
            cond = False

loop = asyncio.new_event_loop()
loop.create_task(flash_leds())
loop.create_task(get_btns(loop))
loop.run_forever()
loop.close()
led4.write(0x0)
led5.write(0x0)
print("Done.")
```

Lab work

Using the code from previous cell as a template, write a code to start the blinking when button 0 is pushed and stop the blinking when button 1 is pushed.

In [ ]:

```
import asyncio
cond = True
#startstop = True
startstop = False

async def flash_leds():
    global cond, start
    while cond:
        #make an if for when button 0 has been pressed to turn on the led flashing
        if startstop:
            led4.write(0x1)
            led5.write(0x7)
            await asyncio.sleep(0.1)
            led4.write(0x0)
            led5.write(0x0)
            await asyncio.sleep(0.05)
            led4.write(0x1)
            led5.write(0x7)
            await asyncio.sleep(0.1)
            led4.write(0x0)
            led5.write(0x0)
            await asyncio.sleep(0.05)

            led4.write(0x7)
            led5.write(0x4)
            await asyncio.sleep(0.1)
            led4.write(0x0)
            led5.write(0x0)
            await asyncio.sleep(0.05)
            led4.write(0x7)
            led5.write(0x4)
            await asyncio.sleep(0.1)
            led4.write(0x0)
            led5.write(0x0)
            await asyncio.sleep(0.05)
        #Make an else for when button 0 has not been pressed. this will sleep and a
        else:
            await asyncio.sleep(.1)

#Function for button0
async def button0(_loop):
    global cond, start, startstop
    while cond:
        await asyncio.sleep(0.01)
        if base.buttons[0].read() != 0:
            startstop = True

#Function for button1. Note: I could make both buttons in one function (would be be
async def button1(_loop):
    global cond, start, startstop
    while cond:
        await asyncio.sleep(0.01)
        if base.buttons[1].read() !=0:
            #_loop.stop()
                #I have the above as a comment because we want this to run forever!
            startstop = False
```

```
loop = asyncio.new_event_loop()
loop.create_task(flash_leds())
loop.create_task(button0(loop))
loop.create_task(button1(loop))
loop.run_forever()
loop.close()
led4.write(0x0)
led5.write(0x0)
print("Done.")
```

In [ ]: