Interacting with GPIO from MicroBlaze

In [1]:
```
from pynq.overlays.base import BaseOverlay
import time
from datetime import datetime
base = BaseOverlay("base.bit")
```

In [2]:
```
%%microblaze base.PMODB

#include "gpio.h"
#include "pyprintf.h"

//Function to turn on/off a selected pin of PMODB
void write_gpio(unsigned int pin, unsigned int val){
    if (val > 1){
        pyprintf("pin value must be 0 or 1");
    }
    gpio pin_out = gpio_open(pin);
    gpio_set_direction(pin_out, GPIO_OUT);
    gpio_write(pin_out, val);
}

//Function to read the value of a selected pin of PMODB
unsigned int read_gpio(unsigned int pin){
    gpio pin_in = gpio_open(pin);
    gpio_set_direction(pin_in, GPIO_IN);
    return gpio_read(pin_in);
}
```

In [3]:
```
write_gpio(0, 2)
read_gpio(1)
```

Out[3]:
```
pin value must be 0 or 1
1
```

Multi-tasking with MicroBlaze

In [4]:
```
base = BaseOverlay("base.bit")
```

In [5]:
```
%%microblaze base.PMODA

#include "gpio.h"
#include "pyprintf.h"

//Function to turn on/off a selected pin of PMODA
void write_gpio(unsigned int pin, unsigned int val){
    if (val > 1){
        pyprintf("pin value must be 0 or 1");
    }
    gpio pin_out = gpio_open(pin);
```

```
    gpio_set_direction(pin_out, GPIO_OUT);
    gpio_write(pin_out, val);
}

//Function to read the value of a selected pin of PMODA
unsigned int read_gpio(unsigned int pin){
    gpio pin_in = gpio_open(pin);
    gpio_set_direction(pin_in, GPIO_IN);
    return gpio_read(pin_in);
}

//Multitasking the microblaze for a simple function
int add(int a, int b){
    return a + b;
}
```

In [6]:
```
val = 1
write_gpio(0, val)
read_gpio(1)
```

Out[6]:
```
1
```

In [7]:
```
add(2, 30)
```

Out[7]:
```
32
```

Lab work

Use the code from the second cell as a template and write a code to use two pins (0 and 1) for send and two pins (2 and 3) for receive. You should be able to send 2bits (0~3) over GPIO. You'll need to hardwire from the send pins to the receive pins.

In [5]:
```
from pynq.overlays.base import BaseOverlay
import time
from datetime import datetime
base = BaseOverlay("base.bit")
```

In [6]:
```
%%microblaze base.PMODB

#include "gpio.h"
#include "pyprintf.h"

int val0 = 0;
int val1 = 0;


int ret  = 0;
int ret2 = 0;
int ret3 = 0;

//Function to turn on/off a selected pin of PMODB
void write_gpio(unsigned int pin0, unsigned int pin1, unsigned int val){
    if (val > 3){
        pyprintf("pin value must be 0 through 3");
    }
```

```
    if (val == 0){
        val0 = 0;
        val1 = 0;
        }
    else if (val == 1){
        val0 = 1;
        val1 = 0;
        }
    else if (val == 2){
        val0 = 0;
        val1 = 1;
    }
    else if (val == 3){
        val0 = 1;
        val1 = 1;
    }


    gpio pin_out0 = gpio_open(pin0);
    gpio_set_direction(pin_out0, GPIO_OUT);
    gpio_write(pin_out0, val0);

    gpio pin_out1 = gpio_open(pin1);
    gpio_set_direction(pin_out1, GPIO_OUT);
    gpio_write(pin_out1, val1);
}


unsigned int read_gpio(unsigned int pin2, unsigned int pin3){

    int array[2] = {};


    gpio pin_in2 = gpio_open(pin2);
    gpio_set_direction(pin_in2, GPIO_IN);
    ret2 =  gpio_read(pin_in2);

    pyprintf("ret2: %d\n", ret2);

    gpio pin_in3 = gpio_open(pin3);
    gpio_set_direction(pin_in3, GPIO_IN);
    ret3 = gpio_read(pin_in3);

    pyprintf("ret3: %d\n", ret3);

    array[0]=ret2;
    array[1]=ret3;
    int a[2]={ret2, ret3};

    pyprintf("array: %d%d\n", a[0],a[1]);


    int array0[2] ={0,0};
    int array1[2] ={1,0};
    int array2[2] ={0,1};
    int array3[2] ={1,1};

    if ((a[0] == array0[0]) && (a[1]) == array0[1]){
        ret = 0;
        pyprintf("ret: %d\n", ret);
```

```
        return 0;
    }

    else if ((a[0] == array1[0]) && (a[1]) == array1[1]){
        ret = 1;
        pyprintf("ret: %d\n", ret);
        return 1;
    }

    else if ((a[0] == array2[0]) && (a[1]) == array2[1]){
        ret = 2;
        pyprintf("ret: %d\n", ret);
        return 2;
    }

    else if ((a[0] == array3[0]) && (a[1]) == array3[1]){
        ret = 3;
        pyprintf("ret: %d\n", ret);
        return 3;
    }

    gpio_close(pin2);
    gpio_close(pin3);

    //pyprintf("ret: %d\n", ret);
    return ret;

}
```

In [7]:
```
# After physically wiring pin 0 to pin 2 and pin 1 to pin 3
write_gpio(0, 1, 3)
read_gpio(2, 3)
```

```
ret2: 1
ret3: 1
array: 11
ret: 3
```

Out[7]:
```
3
```

In [8]:
```
#pinout
input_bit0  = 0
input_bit1  = 1

output_bit0 = 2
output_bit1 = 3
```

In [9]:
```
# testing with user-defined inputs. inputted 0
num = input('input number (0:3):')
```

```
input number (0:3):0
```

In [10]:
```
print(num)
num=int(num)
```

```
0
```

In [11]:
```
write_gpio(input_bit0, input_bit1, num)
read_gpio(output_bit0, output_bit1)
```

```
ret2: 0
ret3: 0
array: 00
ret: 0
0
```

Out[11]:

In [ ]: