Final Report For the Team ENGOP

Team Name: ENGOP
Project Name: FPGA controlled Robotic Arm
Team Member: Xiang Gong, Hao Wu, Zhenyang Wu, Ying Shi

1. Project Overview

Our project is about using FPGA to control the robotic arm and a kart. We use an USB keyboard to produce the control signal and use the FPGA to decode the signal from the keyboard. After that, FPGA send the decoded signal to the driver circuit from its PMOD port. The driver circuit could transfer a low level power source into a high level voltage power supply and activate the motors of the robotic arm and kart through it. The robotic arm could be controlled in three different mode. Manual mode could let it move by the user's order. Record mode could record the move the robotic arm just did in 20 seconds and the Auto mode could repeat the movement of the record mode by pressing a button R.

2. Description of the modules

There are 2 modules in out project which are the keyboard module and the controller module. The keyboard module is in charge of translate the signal from a serial input from the USB keyboard and give it to the controller module when a key is released or still being pressed. The controller module is in charge of translate the signal from the keyboard module and send it out through the controller module.

2.1 keyboard module

The keyboard module could get the data from the keyboard by a 1-bit serial input and send the output signal to the controller by a 8-bit data output and a 2 bit mode signal.

```
always@(posedge flag) // Printing data obtained to led
begin
  if (led==8'hff)
       mark=1;
       if(data_curr!=8'hf0)
       begin
            if (mark!=0)
```

```
            begin
                if (data_curr==8'h2d)//r for auto mode
                begin
                released=1;
                end
                if (data_curr==8'h2c)//t for record mode
                begin
                released=2;
                end
                if (data_curr==8'h35)//y for manual mode
                begin
                released=0;
                end
                led<=data_curr;
            end
            else
                led<=8'hff;
        end
        else
        begin
            mark=0;
        end
End
```

By adding a flag to keyboard controller, we can send data to motor controller by keeping pressing a key(for manual mode) or pressing a key once(for auto mode).
1. mark=1 means last key is released and reset data to 8h'ff to receive new input key.
2. mark= 0 means the key is still being pressed and output data cannot be changed.
By applying these code, we could choose the mode by pressing the button R, T or Y if the button is released and give orders to the robotic arm by keep pressing other buttons.

2.2 controller module

The controller module can get the translated 8-bit keyboard message and a 2-bit mode signal from the keyboard module. The signals will be translate into several 1-bit output to the PMOD to activate the motors. Each motor is controlled by 2 outputs which represent forward and backward of the motor. If both two output are 0 the motor will stop working.
First, it will generate a much slower clk by dividing the original clock signal and make a new clk with a 0.2 second clock period. Since the clock period is 0.2s, the statues of the motor will change in every 0.2s if any button is pressed or released.
It has three different mode for controlling the robotic arm, which are manual mode, record mode and auto mode. The module will check the value of the mode signal(which is called released) at every positive edge of the new clock signal and make a specific translation according the signal.

2.2.1 manual mode

If button Y is pressed or no mode controlling button is pressed, it will enter the manual mode. It will check if button w, s a, d,f g, h, n, j or m is pressed and it will assign values to the PMOD output for activating the motors. It can also control the kart in this mode by pressing arrow up, down, left and right on the keyboard.

2.2.2 Record mode

First we made a 3- dimension array as the memory of the instructions the user made to the robotic arm. The three dimensions stand for the current value of the time(PC), the number of the motor and the state of the motor.

After a button T is pressed, first it will clear the memory and reset all the flags to the original values. Then it will check if any button is pressed just like the manual mode. If a useful button is pressed, it will first assign values to the motor outputs and then record the information of this instruction in the memory. The PC will be increased by 1 in every positive edge of the clock and a register called End of the Record(EndOfRec) which represents the ending time of the record will also be increased by 1.

2.2.3 Auto mode

If a button R is pressed it will enter the auto mode. It could repeat the movement the robotic arm just did in 20s. Since the robotic arm has already been moved to another position in the record mode, it should be moved back to its original position in the auto mode before it repeats the instructions. So every time an R is pressed, the memory will be read backward at first by minus 1 from the PC until it reaches zero. We also assign the value to the output of the motors from the register for the opposite direction when the PC is decreasing to let it move in the opposite direction and move back to the original position.

When the PC drop to zero the flag will be changed and it will start increasing until it reaches the end of the record or 200. The output could get the direct value from the memory at this time. So the robotic arm will move just the way the user did in the record mode.