VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ADVANCED MICROCONTROLLERS – ECE4002 DIGITAL ASSIGNMENT

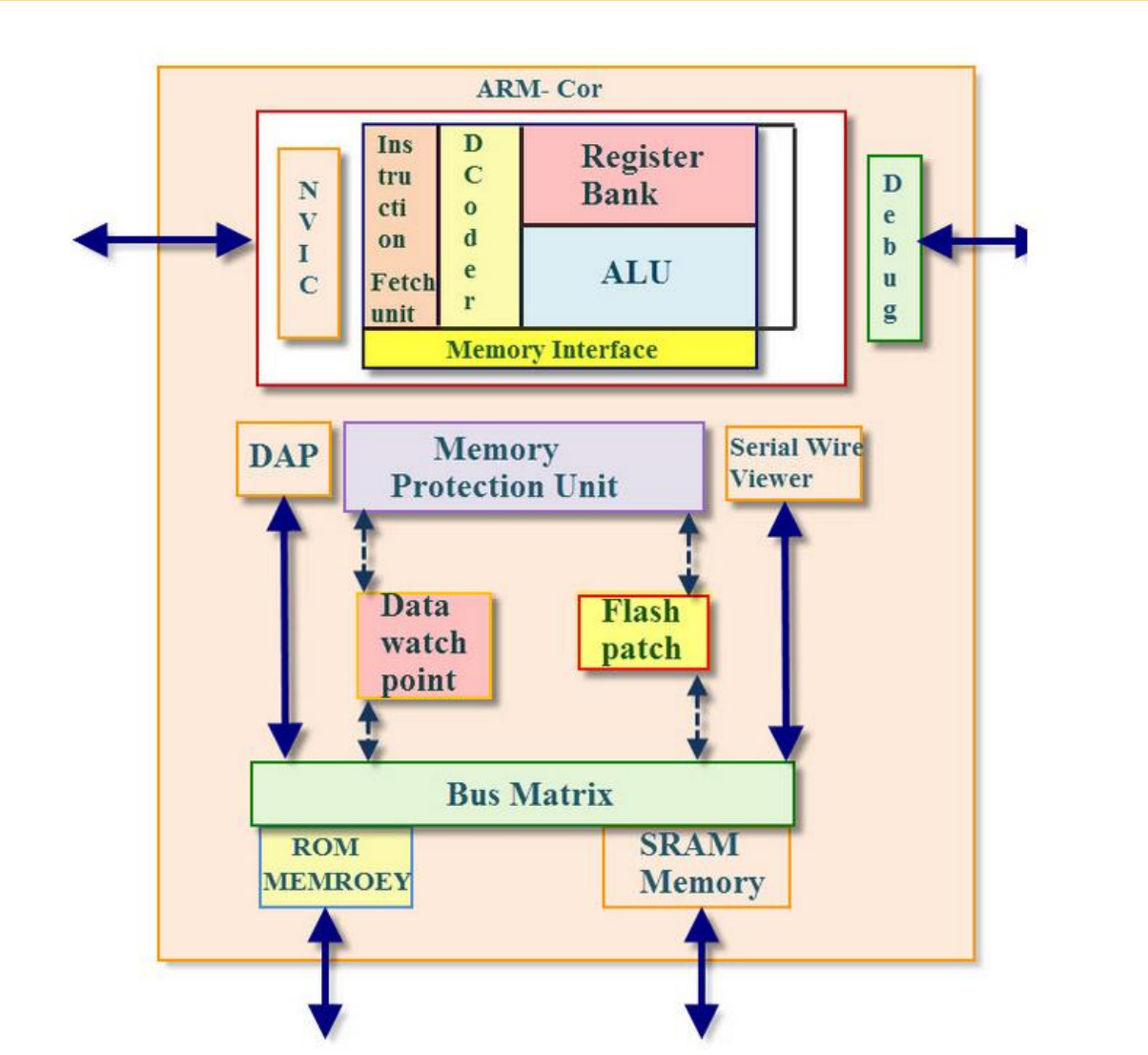
18BIS0043 – Allen Ben Philipose

INDEX

This report, titled **ARM CONTROLLER PORT PROGRAMMING** is submitted to **Mr. VIDHYAPATHI C.M,** Assistant Professor (Senior), Department of Embedded Technology, School of Electronics Engineering, VIT, after completing the assignment of **ECE4002: Advanced Microcontrollers,** conducted during **slots: B2+TB2,** to the best of my knowledge

LPC1768

A high-performance 32-bit unit, the cortex-M3 ARM chipset, offers the development partners significant advantages. The processor follows "Harvard architecture" that includes numerous instructional and data buses to connect with the ROM and RAM memory.



Cortex-M3 has an amazing 32-bit performance, especially considering its incredibly low power consumption. It is usable without any fee for licensing and is widespread among many smart home appliances and other embedded applications.

PORT DESCRIPTION

The General-Purpose Input/Output (GPIO) determines the direction of the pins and the following registers determine various aspects of port management.

FIOSET pins of the selected port are changed to high at the time of output. This is essentially a set function.

FIOCLR pins of the port is changed to low at the time of output. This is essentially a clear function.

FIOPIN also called as pin value register since it saves the current values of the pins.

FIODIR register stores each Input/Output port's direction assignments. It displays 0 for input and 1 for output. It is also called direction control register.

FIOMASK register gives the allowance for the access of each Input/Output port bit using the functions set and clear. If 1, access is denied, if 0, access is approved. Also called as the mask register.

The most common IC packaging is called "LQFP100", which contains a hundred pins. Among them, 30 are for Main Power Supply, abbreviated as V_{DD} and V_{SS}, the crystal oscillator for RTC and processor, abbreviated as XTALs and RTCXs. They also contain some reference debug interface and voltage pins.

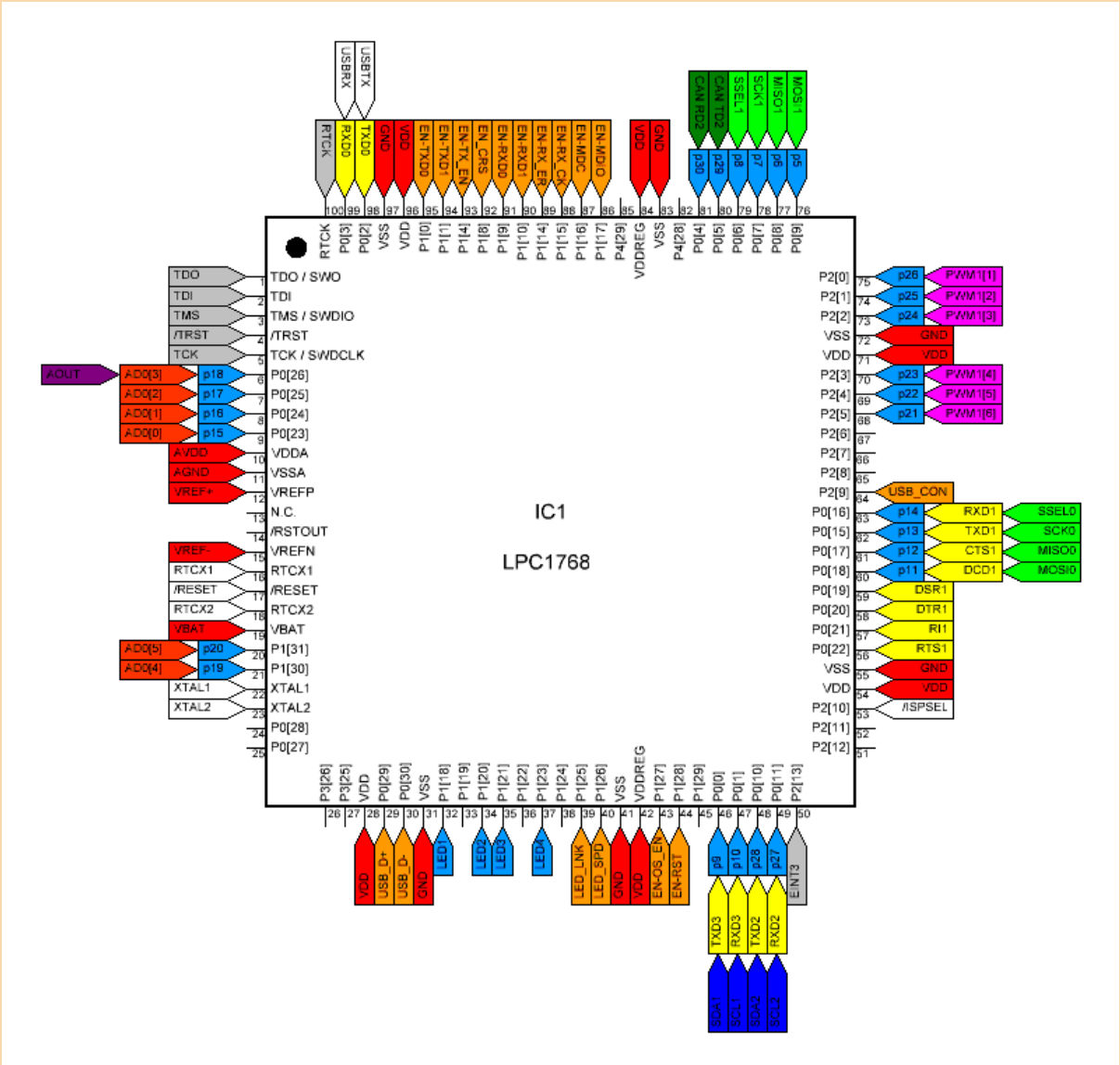
The 70 pins remaining will be used as GPIO pins. Assume a situation to produce a PWM via a pin by using it to connect with the UART?

While these 70 pins are means of the MCU communicating with the outer environment (by GPIO or by any specific device), a procedure is needed for deciding which pin operates as the GPIO pin and which as PWM output pin.

Some values in the registers gives a pre-determined functionality to the program.

Enabling PINSEL0 activates P0[15:0]; PINSEL1 -> P0[31:16]; PINSEL2 -> P1[15:0] with ethernet capabilities; PINSEL3 -> P1[31:16]; PINSEL4 -> P2[15:0]; PINSEL7 -> P3[31:16]; PINSEL9 -> P4[31:16]; and PINSEL10 implies that the Trace port can be enabled.

Value	Function	Short Form
00	Default function	PINSEL_FUNC_0
01	First alternate function	PINSEL_FUNC_1
10	Second alternate function	PINSEL_FUNC_2
11	Third alternate function	PINSEL_FUNC_3



PORT PROGRAMMING

Program to demonstrate the LED blinking at Port 1, 1000 times.

```
#include <lpc17xx.h>
void loop(unsigned int a) {
  unsigned int x,y;
  for(x=0;i<a;x++){
    for(y=0;y<1000;y++);
  }
}
void main() {
    SystemInit();
    LPC_PINCON->PINSEL0 = 0x000000;
    LPC_GPIO0->FIODIR = 0xffffffff;
    unsigned int b;
  for(b=1;b<1001;b++) {
       loop(1000);
       LPC_GPIO0->FIOSET = 0xffffffff;
       loop(1000);
       LPC_GPIO0->FIOCLR = 0xffffffff;
    }
}
```

PINSEL0 activates GPIO0. FIODIR makes Port 0 as an output. FIOSET and FIOCLR are set and clear functions respectively, creating a blinking output at fixed intervals for 1000 iterations.

LED corresponding to switches

```
#include <lpc17xx.h>
#define switch 8
#define led 1
void main() {
    uint32_t st;
    SystemInit();
    LPC_PINCON->PINSEL0 = 0x000000;
    LPC_GPIO0->FIODIR = ((1<<led) | (0<<switch));
  while(1) {
       st = (LPC_GPIO0->FIOPIN>>switch) & 0x01 ;
       if(st==1) {
         LPC_GPIO0->FIOPIN = (1<<led);
       }
       else {
            LPC_GPIO2->FIOPIN = (0<<led);
       }
    }
}
```

The same program displayed earlier is modified to work with a switch instead of blinking LEDs are a regular interval. Port 0 is being used for connecting the LED and the switch at pins 1 and 8 respectively

The codes are based on C programming language

REFERENCES

- https://www.electronicwings.com/arm7/lpc2148-32-bit-arm7tdmi-s-processor-gpio-ports-and-registers
- https://www.electronicshub.org/arm-gpio-introduction/
- https://www.elprocus.com/arm7-based-lpc2148-microcontroller-pin-configuration/
- https://binaryupdates.com/introduction-to-arm7-lpc2148-microcontroller/
- http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C6_MicrocontrollerPorts.htm
- https://www.youtube.com/watch?v=BXXbmqezlwM
- https://www.embedded.com/basics-of-porting-c-code-to-and-between-arm-cpus-from-8-16-bit-mcus-to-cortex-m0/
- https://stackoverflow.com/questions/45493538/how-to-use-gpio-port-control-gpiopctl-in-arm-cortex-m4-tm4c123g-microcontrolle
- https://www.sciencedirect.com/topics/engineering/port-register
- https://www.edgefx.in/arm-microcontroller-architecture-and-its-programming/
- https://sites.google.com/site/coolembeddedlaboratory/home/arm/arm---step-by-step/introduction-to-gpio-pins
- https://www.arm.com/products/silicon-ip-cpu
- http://scmrtos.sourceforge.net/Cortex-M3_Ports
- https://researchdesignlab.com/lpc1768-arm-cortex-m3-development-board.html
- https://www.electronicshub.org/how-to-program-gpio-in-lpc1768/
- https://www.electronicshub.org/getting-started-with-lpc1768/
- https://www.mouser.in/new/stmicroelectronics/stm32f2/
- https://www.exploreembedded.com/wiki/LPC1768:_Switch_and_LED
- https://www.keil.com/dd/vtr/4868/13882.htm
- https://www.exploreembedded.com/wiki/LPC1768:_Register_Configuration(lpc17xx.h)