

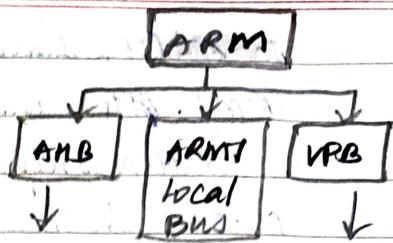
## LECTURE 17 - NXP LPC2148 ARCHITECTURE

LPC2148 is a microcontroller having ARM11TDMI5 as the ARM core and manufactured by NXP, previously known as Philips.

### \* TERMINOLOGIES :

- ① AMBA AHB (Advanced High-Performance Bus) = Operates on Higher Speed Compared to VPB
- ② VPB (VLSI peripheral bus)

### (3) Vectorized Interrupt Controller (VIC)-



Used for

Used for

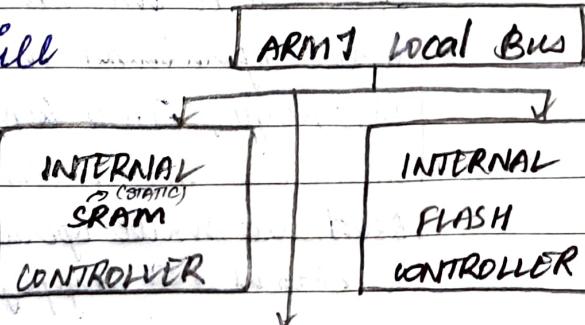
- (4) AHB to VPB Bridge - Converts the connecting all data from low speed devices connected to VPB to a format that is accepted by the AHB. Speed mismatch will be handled.

connecting all peripherals such as USB DMA, IIC, etc. such as ADC, Timer, etc.

- (5) AHB Bridge - Converts data from AHB Bus format to the ARM<sub>7</sub>. This speed mismatch will be handled by AHB Bridge.

- (6) ARM<sub>7</sub> Local Bus - This is the fastest bus available to ARM. This will be operating on the highest speed since it is directly connected and interacting with the processor. Therefore, on comparing other buses in terms of speed, we find that ARM<sub>7</sub> Local Bus > AHB Bus > VPB Bus.

- \* NOTE: Fast General Purpose I/O will be connected to ARM<sub>7</sub> Local Bus where normal General Purpose I/O will be connected to VPB Bus.



Random Access Memory (RAM) is used for data handling. For example, when adding two numbers, the resultant sum will be stored in RAM. Whereas, the program

FAST GENERAL PURPOSE I/O
--------------------------

Special peripheral

2.5x-faster than normal GPIO

Instruction will be stored in flash. depending upon the serial number, we have different bytes of SRAM and FLASH.

In earlier times, programs were stored in Read-Only Memory (ROM). After that PROM was introduced where the program can be written only once. The writing process was done in the factory itself. Later, <sup>(ERASE)</sup> EEPROM was invented where the program can be erased and re-written. The erasing process was done using UV-rays, through a transparent window present in the center of IC, rays were passed to the silicon chips inside the ROM. This window will be covered by the company's logo sticker to avoid light from corrupting the data inside.

The above process had a drawback. A separate UV eraser was required for erasing the program. Moreover, it took 20 minutes to erase the data and only 1000 erase cycles can be performed. To overcome these problems, E<sup>2</sup>PROM was developed.

Electrically Erasable Programmable Read-Only Memory (EEPROM). Here, there was no need for a separate UV eraser, electrical signals were used. But again, there was a trouble. Data can be written only serially here, bulk read and write operations were not possible. In order to have this advantage, FLASH ROM was developed.

Read and write operations can be performed in FLASH with a normal +5V supply and erase cycles count upto 1,00,000 times. This is the latest development that we find in today's pen drives and hard disks.

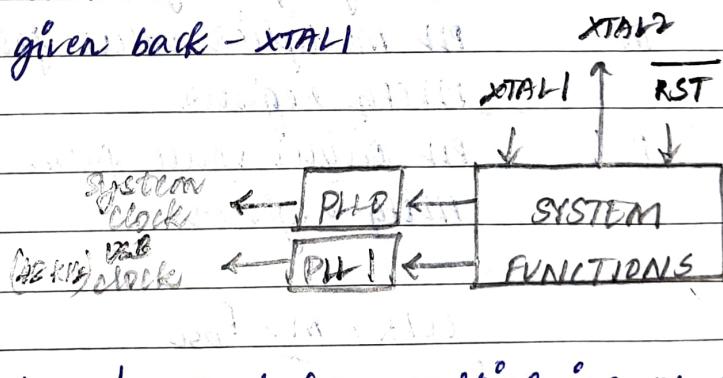
- ① System Functions Peripheral - Handle the clock input given to the controller, a complex digital device / synchronous digital device.

Trainer kit is used to generate clock in a digital lab. 555 astable multivibrator is used to develop this clock. But, accurate clocks are not generated here owing to many passive components such as resistors, capacitors, etc. connected to it, thereby have their own tolerance and varying the final clock output.

- ② Crystal oscillator - This is used to generate a clock in a microcontroller / microprocessor. This is connected to XTAL1 and XTAL2 of the system functions. It cannot generate its own clock but a small frequency clock is given to the crystal from the microcontroller. This induces a vibration in the crystal in its fundamental frequency (say, 12MHz).

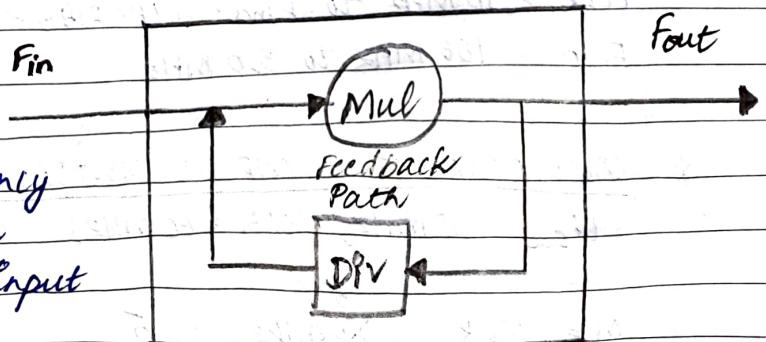
clock In from the controller - XTAL2

crystal frequency given back - XTAL1



- ③ PLL (Phase Locked Loop) - used for multiplying or dividing the clock. System clock is connected to all the peripherals and processors.

PLL is a closed loop control system to generate high frequency by multiplying with given factor to the input frequency.



Basic PLL Block Diagram

In LPC2148 microcontroller, there are 2 PLLs which provide programmable frequencies to the CPU and USB system. The input clock frequency to PLL0 and PLL1 is in the range of 10 MHz to 25 MHz only. It is multiplied up the range of 10 MHz to 60 MHz for CLK and 48 MHz for the USB clock using current controlled oscillator (CCO).

#### \* PLL FREQUENCY CALCULATION (CCLK) :

Table - elements determining PLL's frequency

Element	Description
Frequency of oscillator	$F_{osc}$ the frequency from the crystal oscillator/external oscillator
Core ARM clock	$F_{CCO}$ the frequency of the PLL current controlled oscillator
Multiplication Factor	PLL Multiplier value from the MSEL bits in the PLLCFG register
Division Factor	PLL Divider value from the PSEL bits in the PLLCFG register

$$CCLK = M \times F_{osc}$$

$$F_{CCO} = CCLK \times 2^{XP}$$

$$F_{osc} = 10 \text{ MHz to } 25 \text{ MHz}$$

$$CCLK = 10 \text{ MHz to } F_{max} (\text{LPC2148} - 60 \text{ MHz})$$

$$F_{CCO} = 156 \text{ MHz to } 320 \text{ MHz}$$

#### \* CALCULATION OF M AND P (DIVIDER) VALUE :

$$(F_{osc} = 12 \text{ MHz}, CCLK = 60 \text{ MHz})$$

$$M = \frac{CCLK}{F_{osc}} = \frac{60 \text{ MHz}}{12 \text{ MHz}} = 5$$

Range of  $F_{CO}$  is 156 MHz to 320 MHz.

Substituting  $F_{CO} = 156 \text{ MHz}$ ,

$$P = \frac{156 \text{ MHz}}{2 \times 60 \text{ MHz}} = 1.3$$

Substituting  $F_{CO} = 320 \text{ MHz}$ ,

$$P = \frac{320 \text{ MHz}}{2 \times 60 \text{ MHz}} = 2.67$$

Since the value of P must be an integer, the integer between 1.3 and 2.67 is 2.

### \* PCLK FREQUENCY CALCULATION:

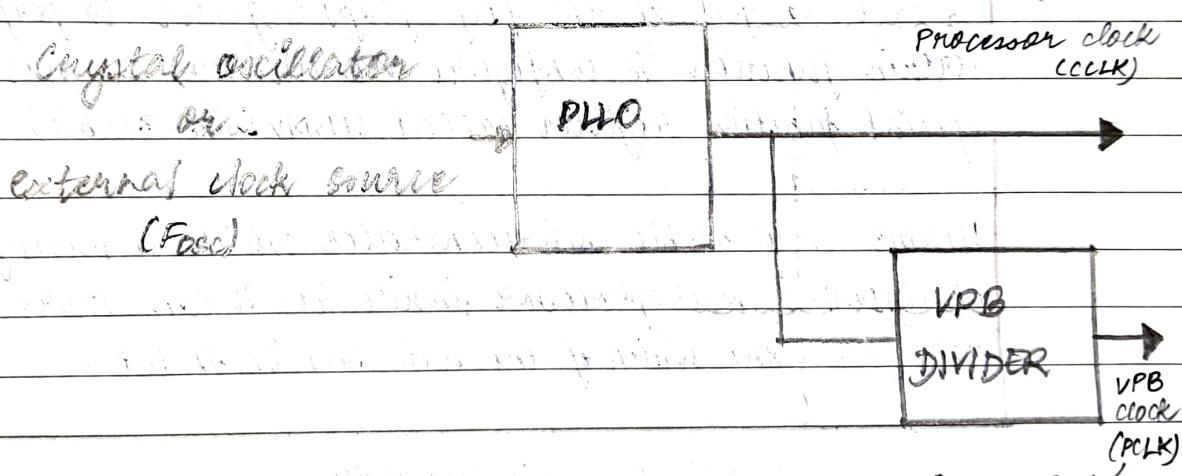


Table - VPB-VLSI Peripheral Bus

Bit	Symbol	Value	Description	Reset value
1:0	VPBDIV	00	VPB bus clock is one fourth of the processor clock. $[60/4 = 15 \text{ MHz}]$	00
		01	VPB bus clock is the same as the processor clock. $[60/1 = 60 \text{ MHz}]$	
		10	VPB bus clock is one half of the processor clock. $[60/2 = 30 \text{ MHz}]$	
		11	Reserved. If this value is written to the VPBDIV register, it has no effect (the previous setting is retained). [DON'T CARE]	

7.2

-

-

Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

Q. How will you configure the VPB divider?

A. To configure a microcontroller, we need to configure some registers (collection of flip flops). Generally in a microcontroller, we have two kinds of registers.

① General Purpose Registers - Store / manipulate data

② Special Function Registers - Configure the peripherals

1 or 0 is fed to the flip-flop register to perform a certain function. To configure VPB divider, we have a special function register called VPBDIV (size - 32-bit).

LPC2148 is a 32-bit microcontroller which is having a ARM11TDMI-S microprocessor inside it. It can store 32-bit of data, i.e., the width of the data bus is 32-bit.

### LECTURE 18 - LED BLINKING

GPIO stands for General Purpose Input Output Peripheral. Similar to ADC peripheral, it is used for interfacing general purpose input or output devices such as an LED (output device) or a switch (input device).

GPIO is a collection of pins, LPC2148 being a 64 pin IC microcontroller contains a package named LQFP - Low-profile quad flat package, meaning there are pins on its four sides and the thickness of the IC is very less.

In general, GPIO is called as GPIO Pouts, i.e., collection of external pins. Through these pins, we interact with the input/output devices.

There are two parts in LPC2148, namely -

- ① PD - 30 Pins
- ② PI - 16 Pins

To refer to an individual pin, we use a terminology as follows-

- ① PD.0 - PD.25
- ② PD.28 - PD.31
- ③ PI.16 - PI.31

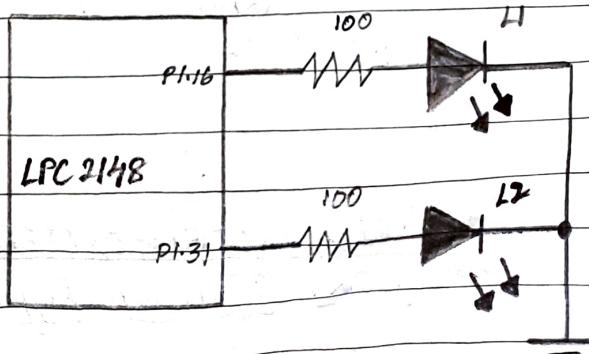
These pins' division is based on multiple criteria such as chip manufacturer, power consumption, size, etc.

Each pin can be given multiple functionality. For example,

46 PD.16 / EINT0 / MATO.2 / CAP0.2  
 ↓      ↓      ↓      ↓  
 GPIO    External    matching    capture  
 PIN 16    Interrupt    pin 20    pin 2

Therefore, four functionalities are multiplexed to a certain pin. This is done to reduce the number of pins, thereby decreasing the size, power consumption, area of cross-section, etc.

### \* LED BLINKING - GPIO



\* NOTE - ARM based microcontroller will be operating on +3.3 V V<sub>CC</sub> (Voltage Common Collector).

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

57

### \* STEPS TO BLINK LEDs :

- ① Configure functionality of P1.31 to P1.16 as GPIO Port  
PINSEL2 = 0 ; → All 32-bits as 0. → LED is an output device
- ② Configure direction of P1.31 to P1.16 as output Port  
IODIR1 = 0xFFFFFFFF; → Digital 1
- ③ Switch on all LEDs on Port1 (Configure Data as Logic High)  
IODET1 = 0xFFFFFFFF; → Digital 0 [Ground]
- ④ Switch off all LEDs on Port1 (Configure Data as Logic Low)  
IDCLR1 = 0xFFFFFFFF;

### Parameters -

+ STEP 1 - PINSEL2 = 0;

① Direction

→ PINSEL Registers - Special Function Register

② Data

PINSEL0 - P0.0 to P0.15

PINSEL1 - P0.16 to P0.31

PINSEL2 - P1.16 to P1.31

→ PINSEL2 Register -

Bit	Symbol	Value	Function	Reset Value
1:0	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
2	GPIO1	0	Pins P1.31-26 are used as GPIO pins.	P1.26/
	DEBUG	1	Pins P1.31-26 are used as Debug port.	
3	GPIO1	0	Pins 1.25-16 are used as GPIO pins.	RTCK
	TRACE	1	Pins 1.25-16 are used as Trace port.	P1.20/
3:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

+ STEP 2 - IODIR1 = 0xFFFFFFFF;

→ IODIR1 (Direction Register) - [Special Function Register]

Bit 31	...	Bit 17	Bit 16	...	Bit 2	Bit 1	Bit 0
--------	-----	--------	--------	-----	-------	-------	-------

↓	↓	↓	↓	...	↓	↓	?
P1.31	...	P1.17	P1.16	...	P1.2	P1.1	P1.0

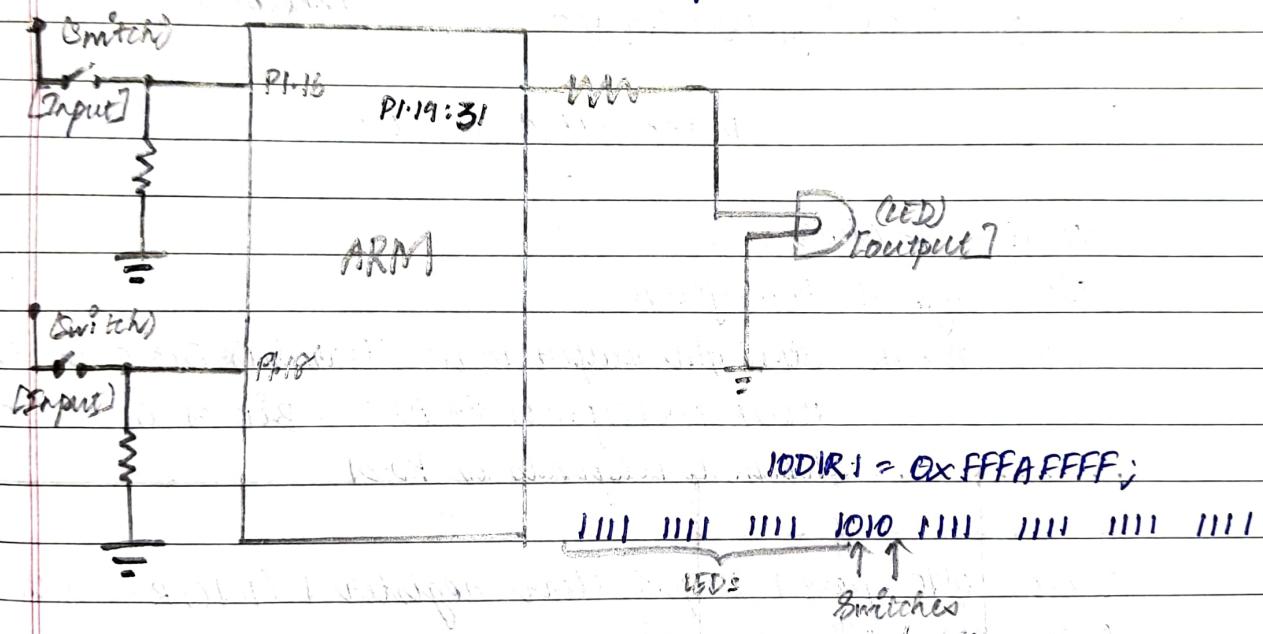
B - Binary  
D - Decimal  
F - Hexadecimal

classmate

Date \_\_\_\_\_  
Page 53

Table - GPIO port 1 Direction register (IODIR - address 0xE002 8018)  
bit description

Bit	Symbol	Value	Description	Reset value
31:0	PIxDIR		Slow GPIO direction control bits. Bit 0 in IODIR controls P1.0 ... Bit 30 in IODIR controls P1.30.	0x0000 0000
		0	Controlled pin is input.	
		1	Controlled pin is output.	



→ STEP 3 - IODIR1 = 0xFFFFFFF;

→ IODET<sub>x</sub> (Data Register - Set) - [Special Function Register]

Bit 31	...	Bit 17	Bit 16	...	Bit 2	Bit 1	Bit 0	??
P1.31	...	P1.17	P1.16	...	P1.2	P1.1	P1.0	

Table: 1 - Set [Logic 1 - Vcc is generated - Pin ON - LED ON]

Bit	Symbol	Description	Reset value
31:0	P0xSET	Slow GPIO output value set bits. Bit 0 in IODET corresponds to P0.0 ... Bit 31 in IODET corresponds to P0.31.	0x0000 0000

Table - GPIO port 1 output set register (IOSET - address 0xED02 8014) bit description

Bit	Symbol	Description	Reset Value
31:0	P1XSET	clear GPIO output value set bits. Bit 0 in IOSET corresponds to P1.0... Bit 31 in IOSET corresponds to P1.31.	0x0000 0000

\* STEP 4 - IOCLR = 0xFFFFFFFF;

→ IOCLR<sub>x</sub> (Data Register - clear) - [Special function register]

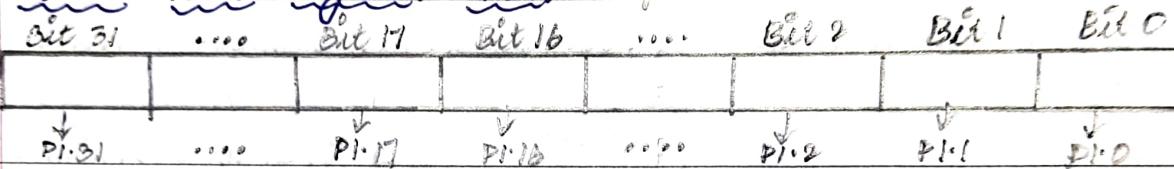


Table 1 - clear [Logic 1] - Pin is cleared - Pin OFF - LED OFF

Bit	Symbol	Description	Reset Value
31:0	P0XCLR	clear GPIO output value Clear bits. Bit 0 in IOCLR corresponds to P0.0... Bit 31 in IOCLR corresponds to P0.31	0x0000 0000

Table - GPIO port 1 output clear register 1 (IOCLR - address 0xED02 801C) bit description

Bit	Symbol	Description	Reset Value
31:0	P1XCLR	clear GPIO output value clear bits. Bit 0 in IOCLR corresponds to P1.0... Bit 31 in IOCLR corresponds to P1.31	0x0000 0000

\* STEPS TO BLINK LEDs :

- ① Configure functionality of P1.31 to P1.16 as GPIO Port  
PINSEL2 = 0;
- ② Configure direction of P1.31 to P1.16 as output port.  
IODIR1 = 0xFFFFFFFF;

\* NOTE: To reduce the brightness of the LED, you may add a resistor.

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

55

### INFINITE LOOP :

- ③ Switch on all LEDs on Port1 [Configure data as logic High]  
IOSET1 = 0xFFFFFFFF;
- ④ Delay [Time taken to execute one cycle is 1/15 seconds]
- ⑤ Switch off all LEDs on Port1 [Configure data as logic Low]  
IOCLR1 = 0xFFFFFFFF;
- ⑥ Delay [TURN OFF → WAIT → TURN ON → WAIT]

### \* REGISTER SUMMARY - GPIO :

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINSEL0	PD.7	PD.6	PD.5	PD.4	PD.3	PD.2	PD.1	PD.0								
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PINSEL0	PD.15	PD.14	PD.13	PD.12	PD.11	PD.10	PD.9	PD.8								
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINSEL1	PD.23	PD.22	PD.21	PD.20	PD.19	PD.18	PD.17	PD.16								
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PINSEL1	PD.31	PD.30	PD.29	PD.28	PD.27	PD.26	PD.25	PD.24								

	15	14	13	12	11	4	3		2		1	0				
PINSEL2							GPIO/TRACE		GPIO/DEBUG		SEL					