Embedded Software

CmpE244

Sept. 29 (Wed) 4:10-8:00pm.

Zoom Link To Be used for the Entire Semester

Harry Li, Office: Eng. Building Rm 269A

E-mail: hua.li@sjsu.edu.

Text messages (650)400-6116.

Grading Policy:

1° Projects & Assignment 30%

2 mandatory Projects, 10% x 2 = 20%

1 Semester-Long Project 10%

2° Midterm: 30%

3° Final: 40%

Organization of the Course

1. CPU Architecture, memory map. Special Purpose Registers for the init & config of Peripheral Controller. Firmware Development. (~3 weeks)

2. Kernel (O.S) Source Distribution IDE (Integrated Development Environment), To be able to optimize Kernel image, to be able to modify existing Device Drivers. to write your own Device Driver. (~3 weeks)

3. Integration & Development of O.S. Kernel + Device Driver + Sensors/Actuators

Stepper Motor Drive
Sensors LSM303
P.I.D controller.
Fourier Transform.
Web Server (GUI)
OpenCV, OpenGL

Introduction

1. Development Setup

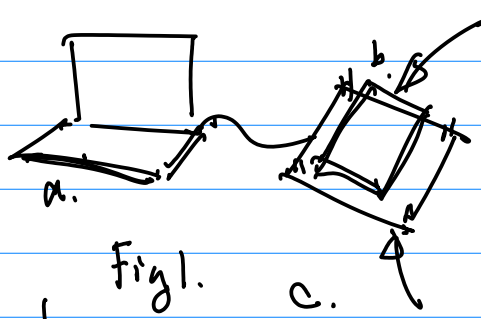


Fig 1.

c. Wire Wrapping Board

a. Host PC/Laptop, Linux Ubuntu 18.04

(Virtual Box Installed, then install Linux on top of it.

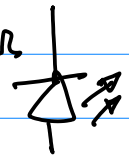
b. Target platform (To Be determined)

c. Wire Wrapping Board

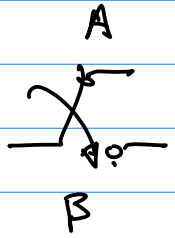
① ~$3\frac{1}{2}$" x 4" physical Dimension Through Holes with metal coating.

(2) 4 mounting Holes @ the corners. 4 stand-offs (Legs)

(3) LED Red/Green Current $\leq 10mA$
Resistors $V_{LED} \cong 1.8 VDC$
200 ~ 500Ω



(4) Toggle switch



3. Selection & Evaluation of A Target platform

Linux O.S. Support

ARM (RISC — Reduced Instruction Set Computer)

- Most Efficient Computation Density Per Unit Power
- Reduced Instruction Set (Smaller collection of machine code instructions → Better Optimization of compiler)
- Forward Looking — Longer Life Cycle

2. O.S. Architecture

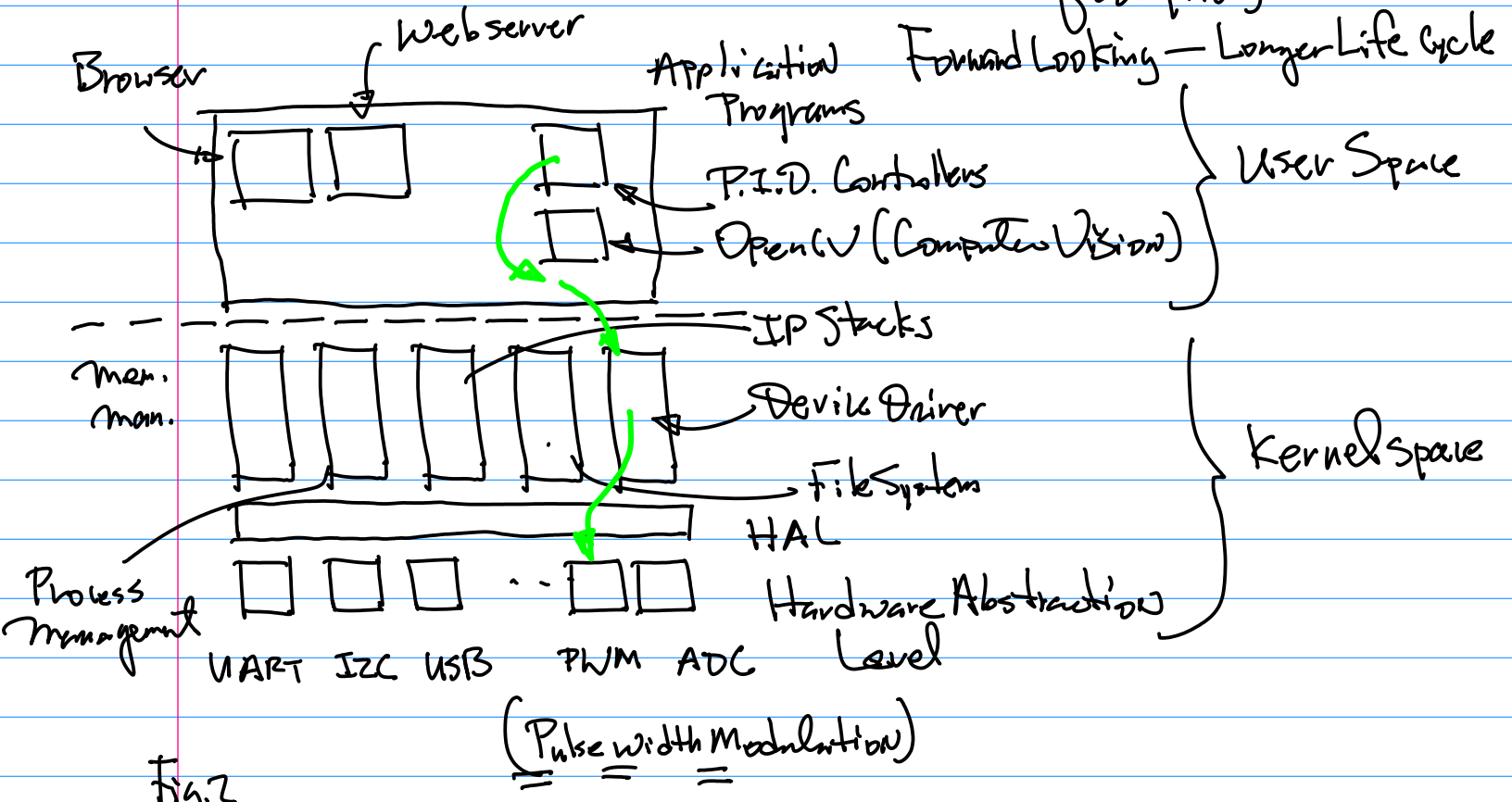


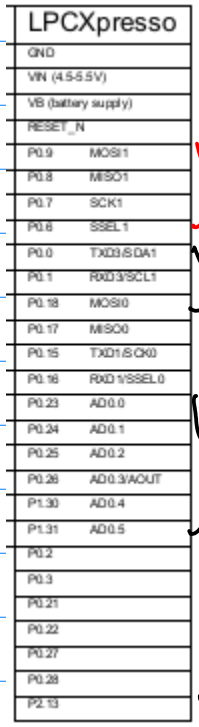
Fig. 2

Target Platforms To consider

1. NXP LPC17xx, 1769
   Clock Rate: 72MHz – 100MHz
   RTOS But Not Unix OS.
   No Linux
   Rich I/O Interface
   Ref: git~2021F-107b- . .

CPU Datasheet —— CPU Architecture well Documented
Memory map well Documented

Example: Sch of LPC1769.



"3+1" pin: SPI1 (Serial Peripheral Interface)

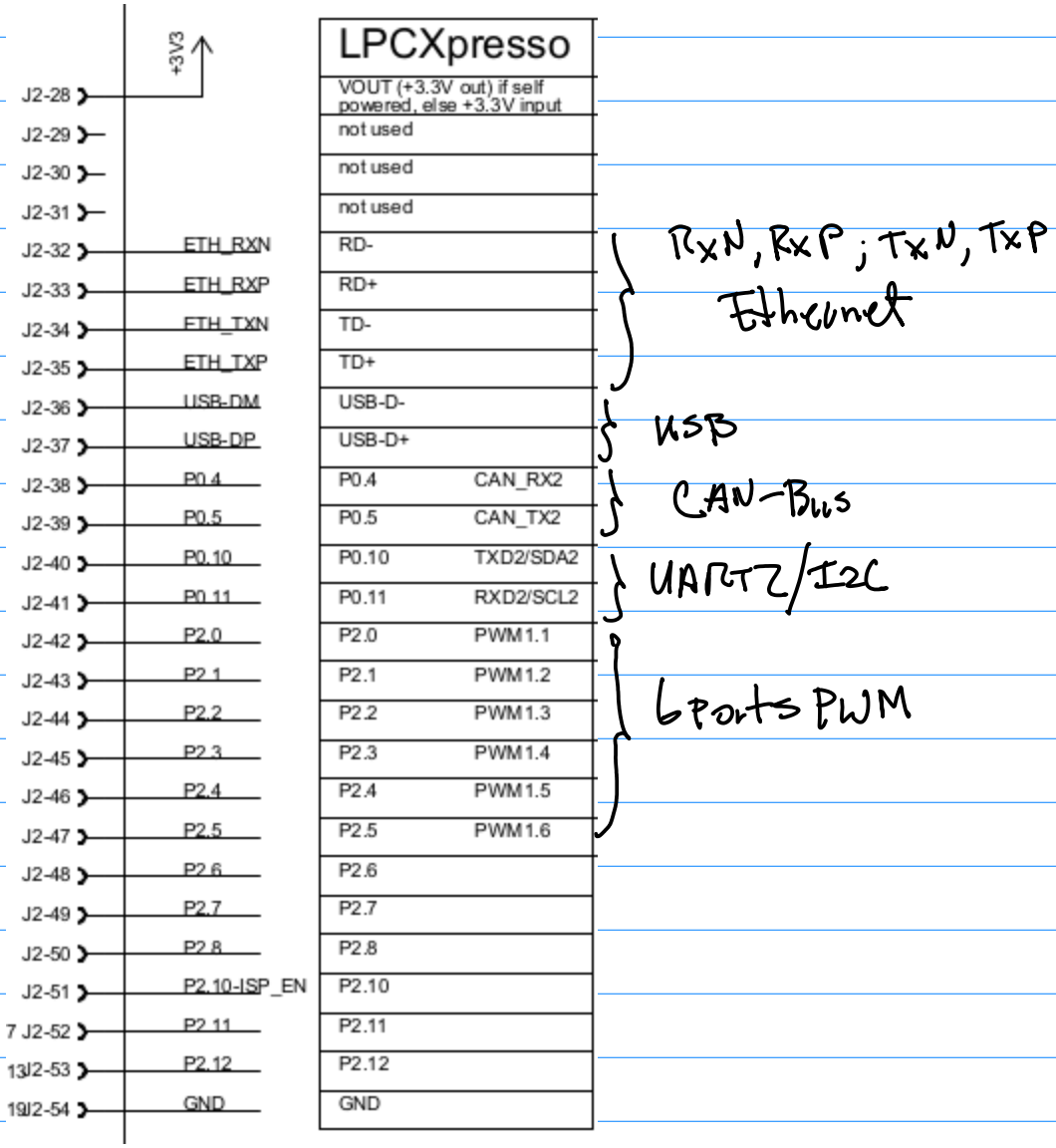
UART3 { Tx Transmission / Rx Receiving } multiplexed

SPI0/UART1

SDA: Serial Data
SCL: Serial CLK
I2C

6 ADC (AD — Analog to Digital Conversion)

GPP (GPIO: General Purpose Port I/O)



Option Target platform: FPGA. Futus Electronics.

FPGA Igloo2: RISC-V.

Superset of ARM Architecture

IP Core: Open Source. Supports RTOS

Limitations: No Unix/Linux O.S.

Smaller Gate Counts. → less

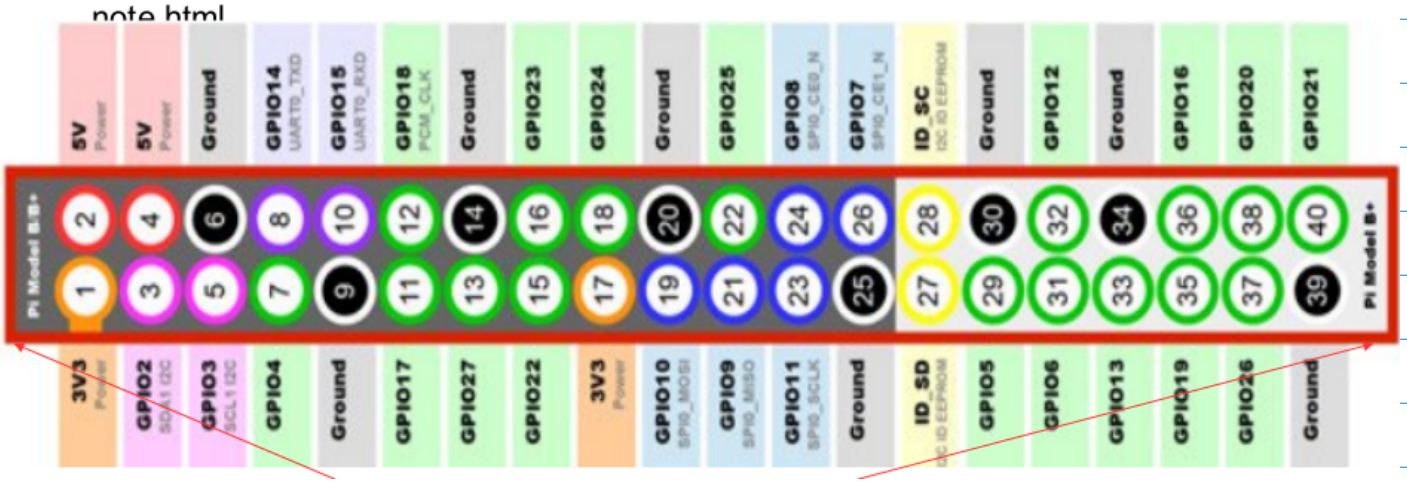
Computational Capability.

Pie. Broadcom BOM

Features: Support Linux/Ubuntu.
Provides machine/Computer Vision Capability; OpenCV.
YoLo (You only Look Once) — Deep Learning, NO ADC

Pie-3 Version B GPIO Pins

https://www.jameco.com/Jameco/workshop/circuitnotes/raspberry-pi-circuit-note.html



Plus: CPU Datasheet → Device Driver Development.
CPU Architecture (memory map, Peripheral Controllers, G.E. (Graphics Engine)

Option: ARM-11 Samsung. CPU: S3C6410x
CPU Datasheet: Architecture Block Diagram — well Document
600~800MHz memory map, well Designed, documented
Support Linux, well Document/Sample code for Driver Development.

State-of-the-Art Feature: Graphics Processing Engine — GPU
LPC17xx, NO; FPGA RISC-V, NO; BCM-Pie G.E. yes
ARM-11. Video Codec, Marginal

Option: NVDA — Jetson TX2 6 CPUs + 256 GPU in a Single Package

CMPE244

Supports Linux/Ubuntu, I/O Interface is limited ; (LPC17xx has the Best I/O I/F Support) ; ~$70 Dev. Kit. Well Documented Datasheet.

Option: NVDA Jetson NANO. Ubuntu Linux Support.
Multiple Cpus + 128 GPU
{ I/O (Limited)
{ Datasheet — Not As Detailed as other platform)

Developer forum is very Active and it gives a good references.

Homework: 1° Form 2-4 person team
By Next week ;
2° Choose Target Platform
3° Bring WireWrapping Board/ Prototype Board to the Class for show & Tell ;
4° Sign up @ Nvidia website as a developer → Kernel (Ubuntu)

{ Source Distribution
{ Jetpack

NXP, MCU Xpresso
www.nxp.com

Sign up as a developer at nxp website
www.nxp.com
MCU Xpresso

Oct 6 (Thursday).
Topics:
1° Architectural Aspects of Embedded System for Software Implementation
Datasheet + Board Sch. + Special Purpose Register + IDE (Compiler and Flash Tool)

Example: LPC1769 ARM Cortex M3
CPU Datasheet
Simpler Architecture
(1) NXP LPC 1769 Baseline
(2) Samsung ARM11 Datasheet
CPU Block Diagram
memory map
Special Purpose Register
Tool Chain / Software Design, Implementation

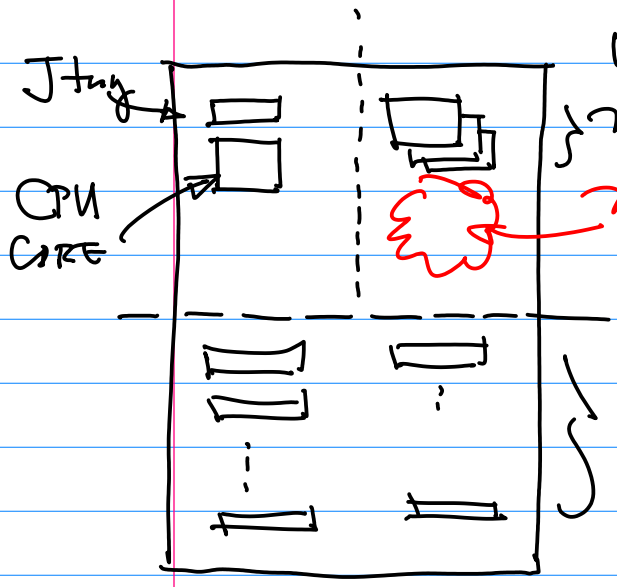


Fig1.

Note: 1. Enumeration of Subsystems is always started as 0.

Vector Graphics Engine
- 2D Graphics
  - a. Display Device Driver
  - b. 2D primitive Graphics Processing
  - c. 2D Transformation
    - line
    - circle
- 3D Graphics

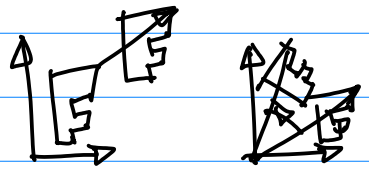
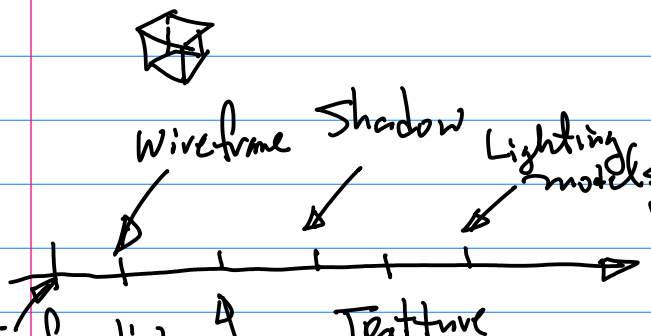
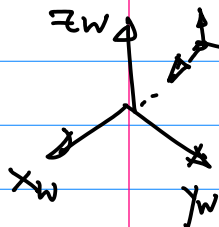


Fig.2b





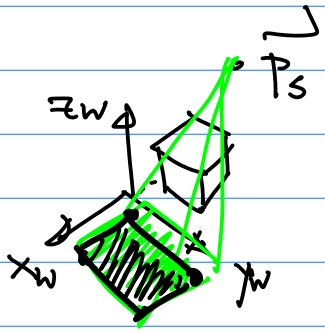


Fig.2

Memory map:

1. RISC: Reduced Instruction Set Architecture

1979 David Patterson
1982 John Hennessy

- Uniformity
- Regularity
- Orthogonality

2. 32 Bit RISC Architecture

Data Bus: 32 bit, Bi-directional

D[m:n] Vector Notation

most Significant Bit — Least Significant Bit

D[31:0]

Endian "Little Endian"

Address Bus: 32 bit, Uni-Directional

ALU (Arithmetic/Logic Unit) 32 bits

Register File: 32 bit

- General Purpose Registers: 32 bit
  - GPRs
- Special Purpose Registers: 32 bit
  - SPRs

GPRs: those that can participate Any meaningful Arithmetic Logic Operations.

SPRs: Those Registers to fulfill special functions, such as init & config. for Peripheral controllers.

3. Memory map

a. Byte Addressable machine

The Smallest memory cell with an unique Address is a Single Byte — "Byte Addressable machine".

Question: What is the max Address for the memory map of a given CPU?

Single 32 Bit Architecture → Addr. Bus 32 bits

$2^{32} = 2^{10} \cdot 2^{10} \cdot 2^{10} \cdot 2^{2}$

2^{10} ... 1K

2^{20} ... 1K x 1K = 1 Meg.

2^{30} ... 1M x 1K = 1 G.

4 GByte (Byte Addressable)

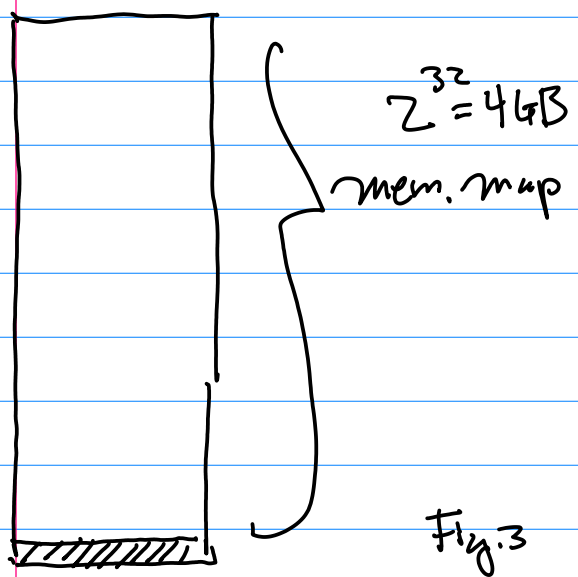


Fig.3

0X0000_0000 System Power_Up Address:

The Address when CPU is powered up, it will go to this memory location to fetch the 1st instruction to execute.

BANKS: A Block of memory.

Divid memory Map into 8 Equal Banks.

1st Bank: BANK 0
2nd Bank: BANK 1
⋮
8th BANK: BANK 7.

Question: How many Address Bits Do we need to define each memory bank? 3 bits

Question: Which 3 bits?

Addr. [31:0]

$a_{31}a_{30}a_{29} \cdots a_2a_1a_0$

Addr[31:29] = $a_{31}a_{30}a_{29}$

Question: Find the Starting Address of Each memory Bank?

Sol:

$a_{31}a_{30}a_{29}$	a_{28}		
0 0 0	0	0X0000_0000	BANK0
0 0 1	0	0X2000_0000	Bank1
0 1 0	0	0X4000_0000	Bank2
⋮			⋮
1 1 1			Bank7

Datasheet, pp.14 from NXP LPC1769 memory map, SSP0 as an example. then, starting Addr. for memory Banks,

Starting Address SSP0:

0X4008_8000

Question: How much memory does SSP0 need?

Peripheral Controllers

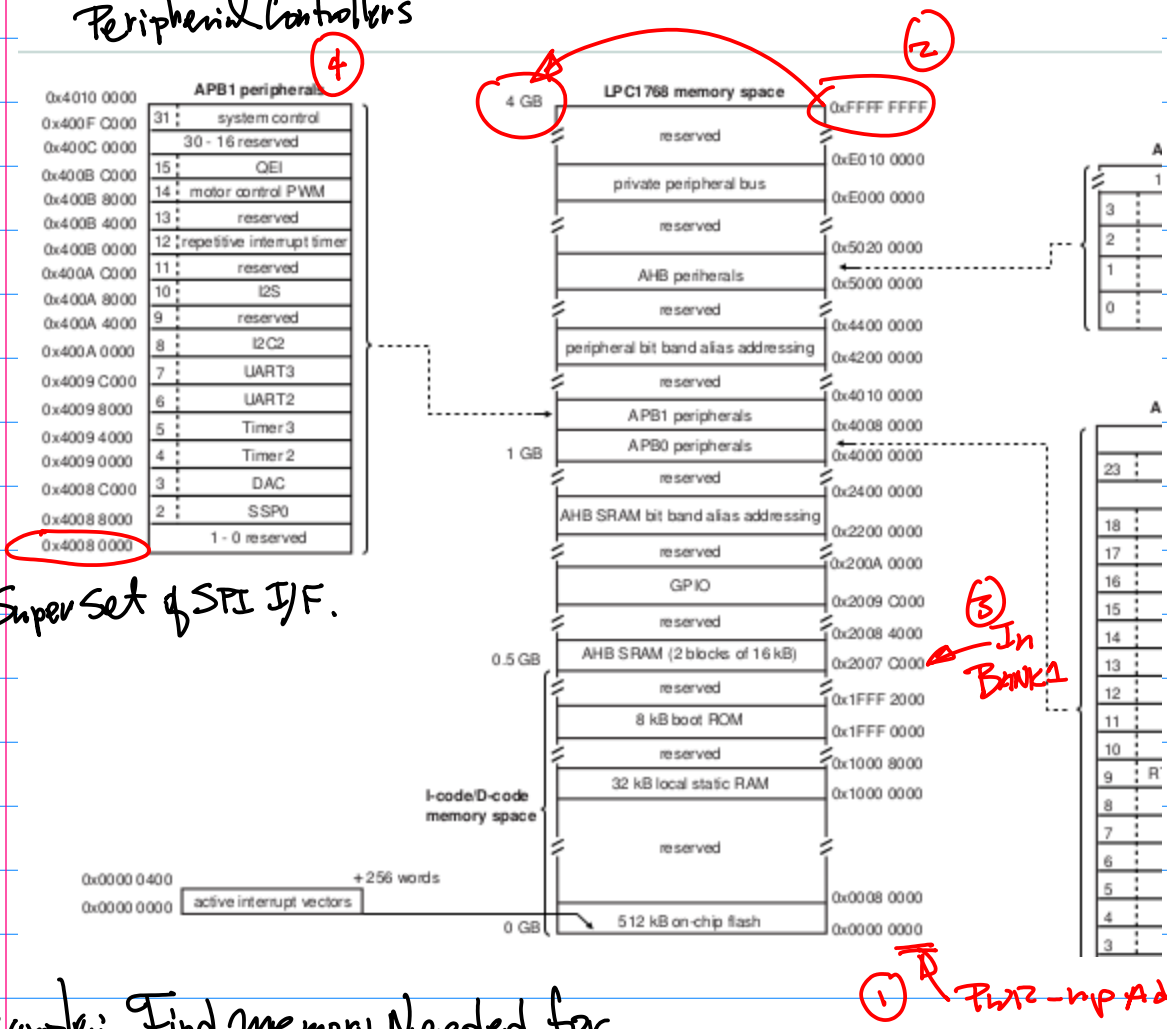


Fig 4

SSP0: Superset of SPI I/F.

Example: Find memory Needed for SSP0.

Starting Addr: 0X4008_8000

End Addr. 0X4008_C000 - 1

= 0X4008_bFFF

8000_bfff Block of memory

For what purpose? Employed By Special Purpose Registers for init & configuration of SSP0,

And to perform Data Input/Output Operation

Special Purpose Registers Design.

focus on GPP (General Purpose Port, eg. GPIO)

3 Common Types of SPR

Control Register : Init & Config.
Data Register : Data I/O Operation
Pull up/Down : Electrical Characteristic of the Controller

4. Naming Convention of Special Purpose Registers.
Let's Design/Define Naming Convention.
follow RISC Design Guidelines

Prefix + Root + Postscript

3 Letters 3 Letters 3 letters

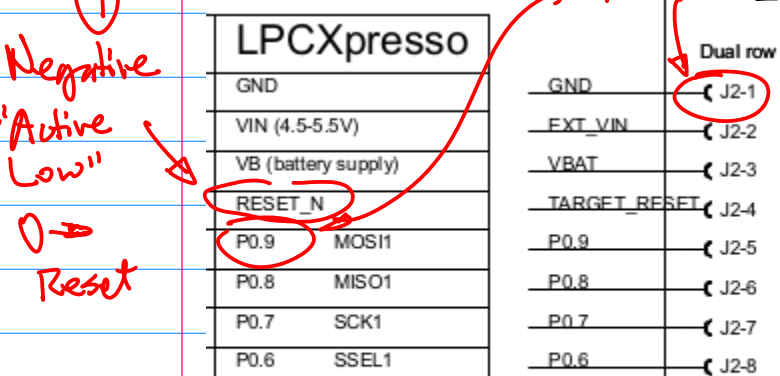
Control Register : GPx CON
GPx : General Purpose Port x
meaning we have more than one general purpose Ports.

C. Typical Number of GPP :
LPC1768 P0, P1, P2, P3.
Samsung Arm 11 :
17 GPPs

S3C6410 includes 187multi-functional input/o

PortName	Number of Pins.	
GPA port	8	UAI
GPB port	7	UAI
GPC port	8	SPI
GPD port	5	PCI
GPE port	5	PCI
GPF port	16	CAI
GPG port	7	SDI
GPH port	10	SDI
GPI port	16	LCI
GPJ port	12	LCI
GPK port	16	Hos
GPL port	15	Hos
GPM port	6	Hos
GPN port	16	EIN
GPO port	16	Mei
GPP port	15	Mei
GPQ port	9	Mei

Fig 5.



a Port (GPP) 0

b Theoretically, pins can be as many as 32.

Fig 6.

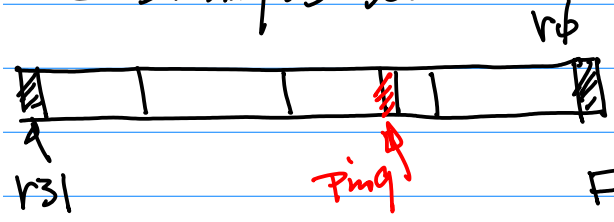


Fig 7

c. GPx DAT ping:

Root

GPxDAT[9]

d. ~~Physical connector pin to~~
GPxDAT[9] is given from
SCH Design. Example, LPC1769

P0.9 → J2-5 (Connector J2, Pin5)

e. GPxPUD (Pull-up/Down)
SPR.

f. Question: How many control functions for a single control Register? Can we have

~~Homework~~: Down Load Image
And Bring up your Target.