

องค์ประกอบของเครื่องคอมพิวเตอร์

และภาษาแอสเซมบลี:

ARM และ RaspberryPi3

บทที่ 6 อุปกรณ์/วงจรอินพุตและเอาท์พุต

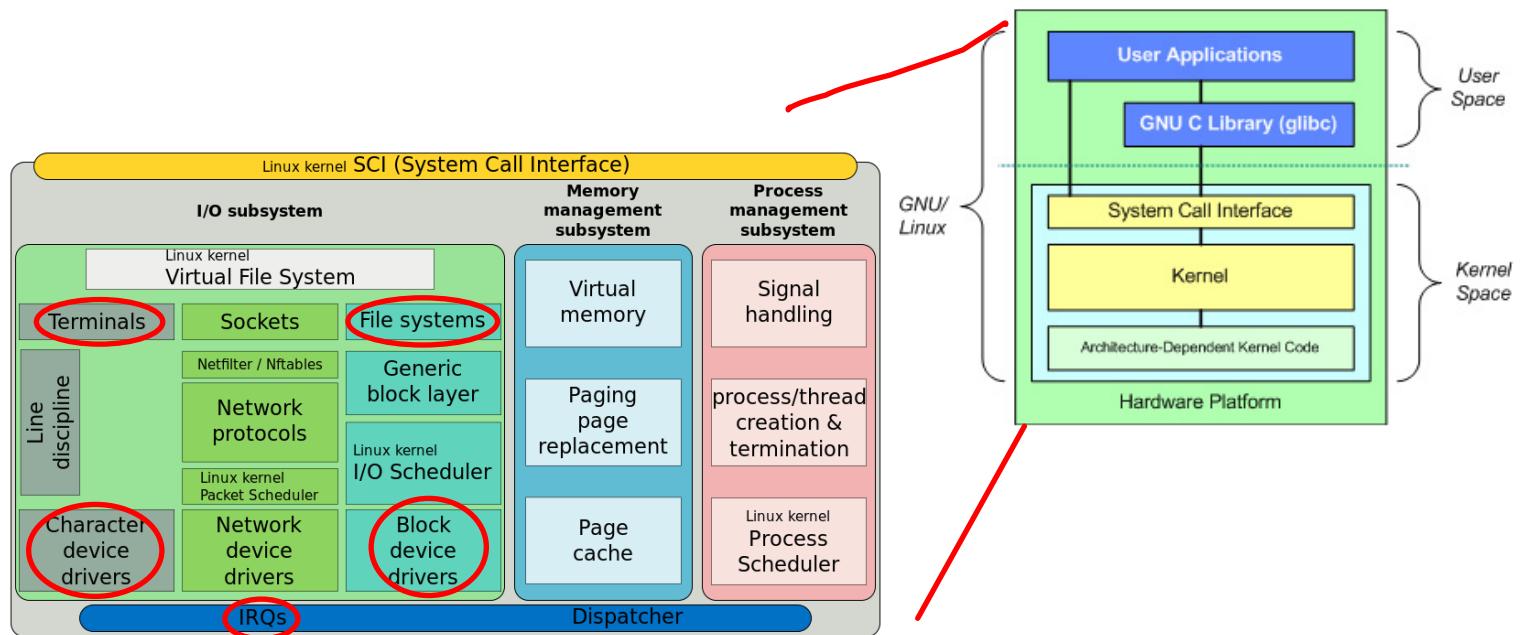
ผศ.ดร.สุรินทร์ กิตติธรกุล

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สารบัญ

- 6.1 สัญญาณ HDMI สำหรับจอภาพ LCD ขนาดใหญ่
- 6.2 สัญญาณ DSI สำหรับจอภาพ LCD ขนาดเล็ก
- 6.3 สัญญาณ CSI สำหรับเชื่อมต่อกล้องขนาดเล็ก
- 6.4 สัญญาณ PCM สำหรับสัญญาณเสียง
- 6.5 สัญญาณภาพและเสียงสำหรับทีวี
- 6.6 สัญญาณ USB 2.0 สำหรับอุปกรณ์ต่อพ่วงต่างๆ
- 6.7 สัญญาณ Ethernet สำหรับสายเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต
- 6.8 สัญญาณ WiFi และ Bluetooth สำหรับการสื่อสารไร้สาย
- 6.9 หลักการ Memory Mapped Input/Output
- 6.10 หัวเชื่อมต่อ 40 ขา (40-Pin Header)
- 6.11 ขา GPIO (General Purpose Input Output)
- 6.12 การขัดจังหวะ (Interrupt)
- 6.13 การเข้าถึงหน่วยความจำโดยตรง (Direct Memory Access)
- 6.14 แหล่งจ่ายไฟ (Power Supply) ของบอร์ด Pi3

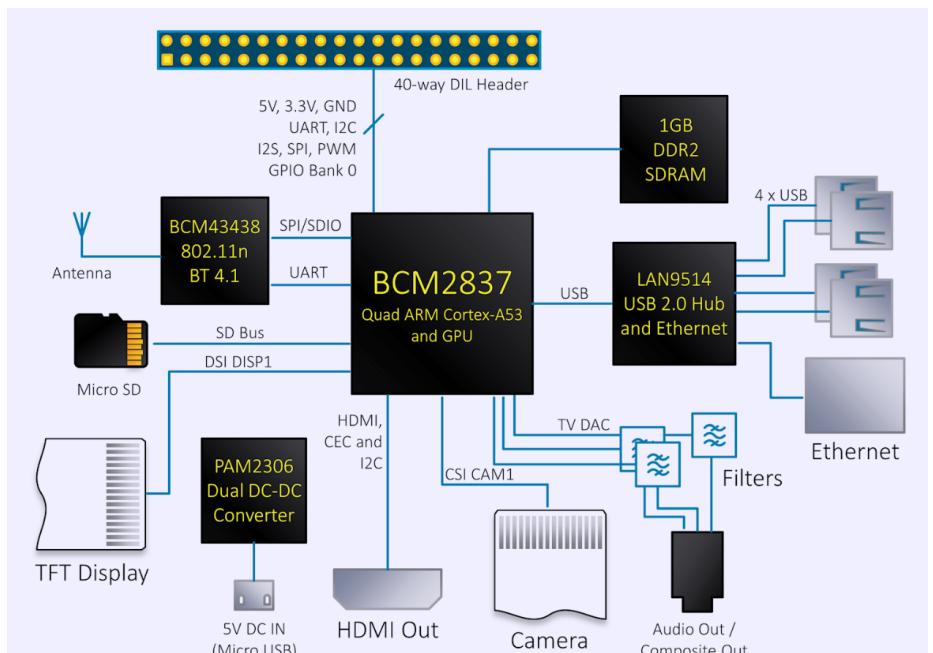
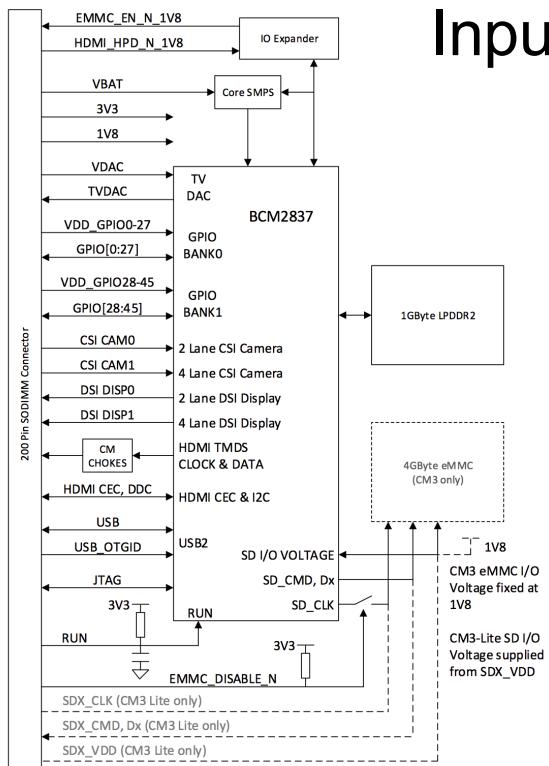
Input/Output: Software Perspective



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

3

Input/Output: Hardware Perspective



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

4

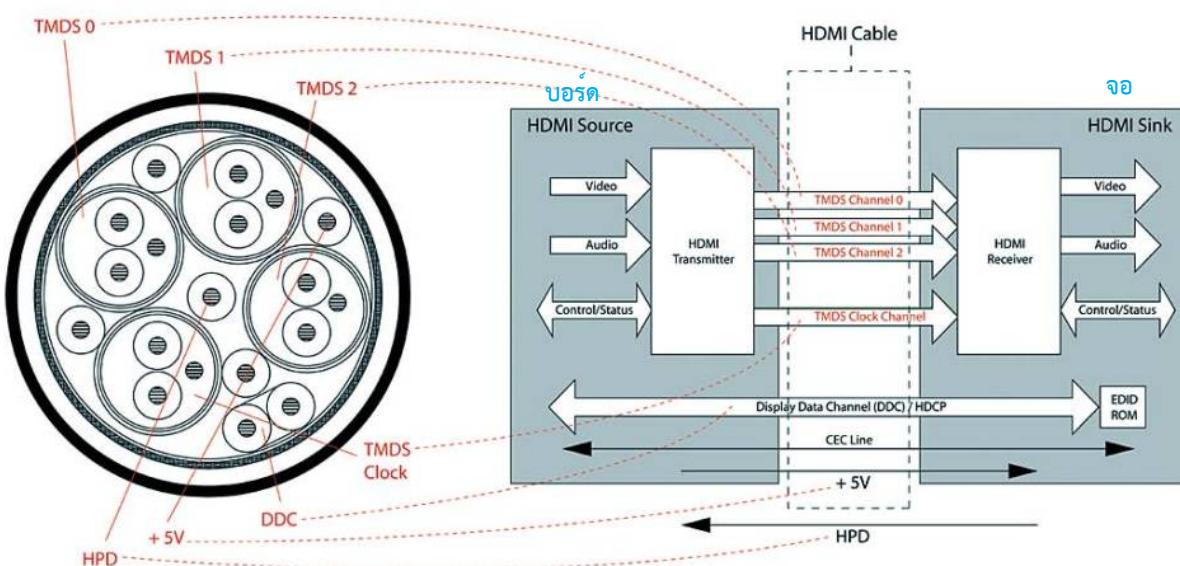
6.1 สัญญาณ HDMI สำหรับจอภาพ LCD ขนาดใหญ่

- การเชื่อมต่อแบบ HDMI เป็นการถ่ายโอนสัญญาณแบบดิจิทัล สามารถส่งได้ทั้งสัญญาณภาพและสัญญาณเสียงไปพร้อมๆ กันด้วยอัตราบิทเรตสูง ระดับจิกะบิทต่อวินาที ความละเอียดภาพที่นิยม คือ 1080×1920 เท่ากับ 1K เส้น
- การเชื่อมต่อด้วยสัญญาณ HDMI เมามะสำหรับการแสดงผลจากเครื่องคอมพิวเตอร์หรือเครื่องเล่นมีเดีย (Media Player) ไปยังจอภาพความละเอียดสูงระดับเอชดี (HD) หรือสูงกว่าสำหรับความละเอียด Ultra HD
- HDMI เวอร์ชันล่าสุด คือ 2.1 ซึ่งพัฒนาเพื่อรับรับวิดีโอที่ละเอียดสูงเฟรม率 8000 เส้นที่ 60 เฟรมต่อวินาที (8K60) และเฟรม率 4000 เส้นที่ 120 เฟรมต่อวินาที (4K120) และเพิ่มถึงเฟรม率 10,000 เส้น (10K) ซึ่งจะทำให้อัตราบิทเรตเพิ่มเป็น 48 จิกะบิทต่อวินาที

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

5

6.1 สัญญาณ HDMI สำหรับจอภาพ LCD ขนาดใหญ่



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

6

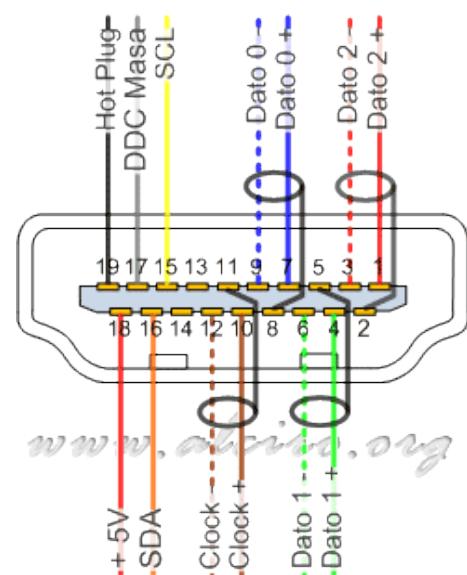
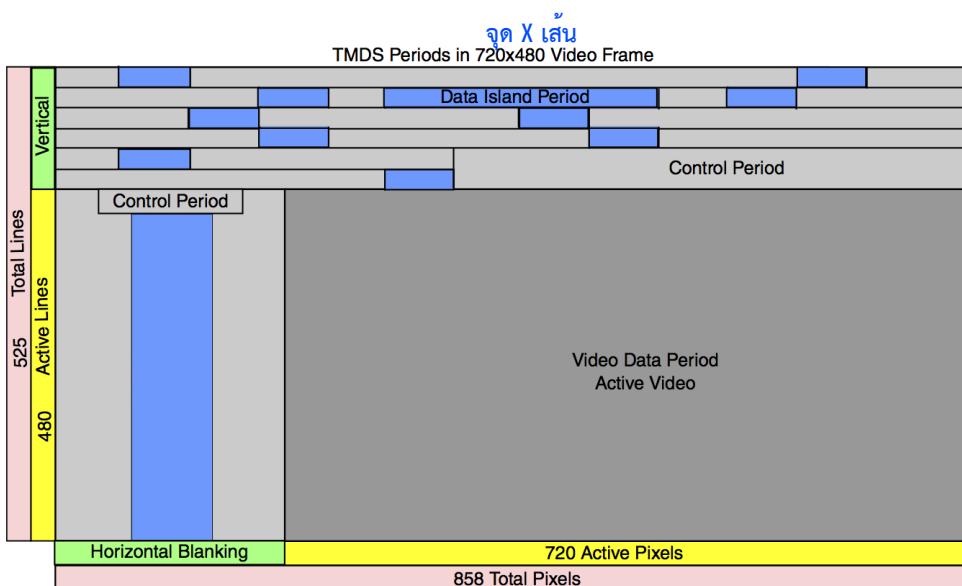
6.1 สัญญาณ HDMI สำหรับจอภาพ LCD ขนาดใหญ่

- สัญญาณ HDMI มีช่องสื่อสาร 5 ช่องแยกจากกัน ได้แก่
 - ช่อง TMDS (Transition-Minimized Differential Signaling) ช่อง TMDS จะส่งข้อมูลภาพวีดีโอ เสียง และข้อมูลเป็นดิจิทัล
 - ช่วงส่งข้อมูลภาพ (Video Data Period) และ
 - ช่วงส่งแพ็กเก็ตควบคุม (Control Period) สำหรับสัญญาณควบคุม เช่น สัญญาณ HSYNC และ VSYNC
- ช่อง DDC (Display Data Channel) ใช้สื่อสารกับเครื่อง Media Player
- ช่อง CEC (Consumer Electronics Control)
- ช่อง ARC (Audio Return Channel)
- ช่อง HEC (HDMI Ethernet Channel)

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

7

6.1 สัญญาณ HDMI สำหรับจอภาพ LCD ขนาดใหญ่



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

8

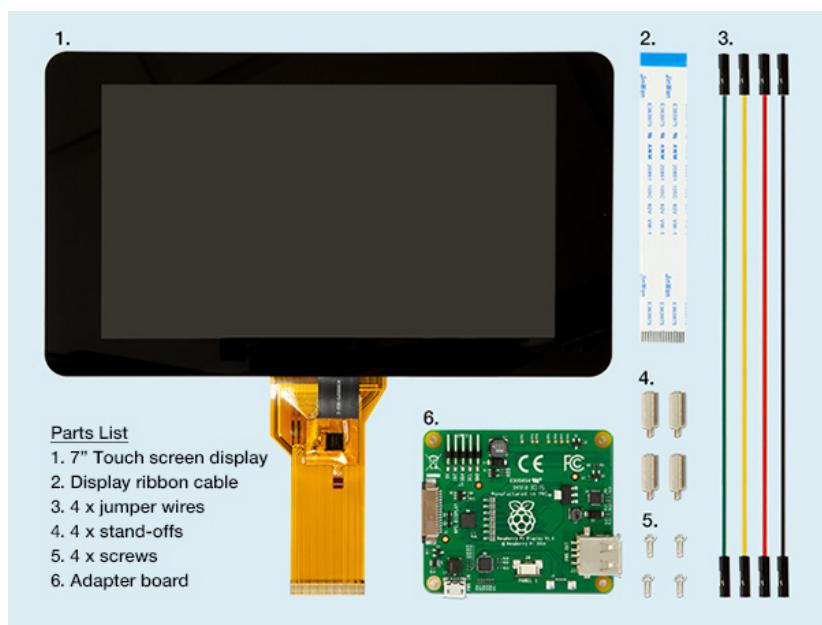
6.2 สัญญาณ DSI สำหรับจอภาพ LCD ขนาดเล็ก

- สัญญาณ DSI (Display Serial Interface) สำหรับเชื่อมต่อกับจอ LCD ขนาดเล็กกับซีพียูบนอุปกรณ์เคลื่อนที่ เช่น โทรศัพท์เคลื่อนที่ แท็บเล็ต คอมพิวเตอร์โน้ตบุ๊ค เป็นต้น เพื่อการแสดงผลในรูปของกราฟิก荷มด
- สัญญาณ DSI นี้ถูกกำหนดเป็นมาตรฐานโดยองค์กรชื่อ MIPI (Mobile Industry Processor Interface) <http://www.mipi.org>
-

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

9

6.2 สัญญาณ DSI สำหรับจอภาพ LCD ขนาดเล็ก



ข่า	ชื่อ	หน้าที่
1	Ground	กราวด์
2	Data Lane 1-	ขาลงเลนข้อมูล 1
3	Data Lane 1+	ขาขึ้นเลนข้อมูล 1
4	Ground	กราวด์
5	Clock N	ขาลงคล็อก
6	Clock P	ขาขึ้นคล็อก
7	Ground	กราวด์
8	Data Lane 0-	ขาลงเลนข้อมูล 0
9	Data Lane 0+	ขาขึ้นเลนข้อมูล 0
10	Ground	กราวด์
11		
12		
13	Ground	กราวด์
14	+3.3 V	ไฟเลี้ยงขนาด 3.3 โวลท์
15	+3.3 V	ไฟเลี้ยงขนาด 3.3 โวลท์

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

10

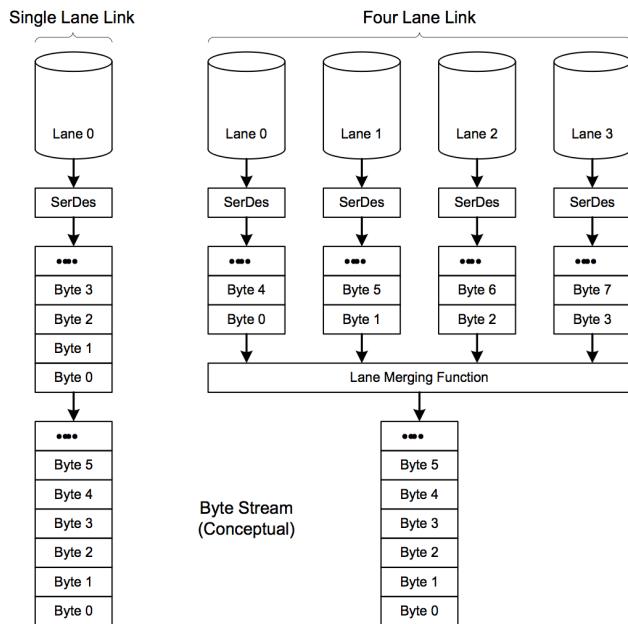
6.2 สัญญาณ DSI สำหรับจอภาพ LCD ขนาดเล็ก

- สัญญาณ DSI แบ่งเป็นชั้นนิดเดียว (Single Lane) และหลายๆ เลนตั้งแต่ 2 เลนขึ้นไป เพื่อกระจายการส่งข้อมูลแต่ละใบที่ข้อมูลไปแต่ละเลน
- ข้อมูลใบที่ 0, 4, 8, ... จะส่งมาทางเลนหมายเลข 0
- ข้อมูลใบที่ 1, 5, 9, ... จะส่งมาทางเลนหมายเลข 1
- ข้อมูลใบที่ 2, 6, 10, ... จะส่งมาทางเลนหมายเลข 2
- ข้อมูลใบที่ 3, 7, 11, ... จะส่งมาทางเลนหมายเลข 3 และสลับกันไปแบบนี้เรื่อยๆ
- การส่งข้อมูลจำนวนหลายๆ เลนพร้อมกันทำได้เร็วขึ้น รองรับการแสดงผลที่ละเอียดมากขึ้น เปลี่ยนแปลงภาพต่อวินาทีได้มากขึ้น การเคลื่อนไหวของภาพจะดีกว่าเดิม
- เมื่อปลายทางรับข้อมูลได้สำเร็จ จะจารับจะนำข้อมูลเหล่านั้นรวมกัน (Lane Merging Function)

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

11

6.2 สัญญาณ DSI สำหรับจอภาพ LCD ขนาดเล็ก

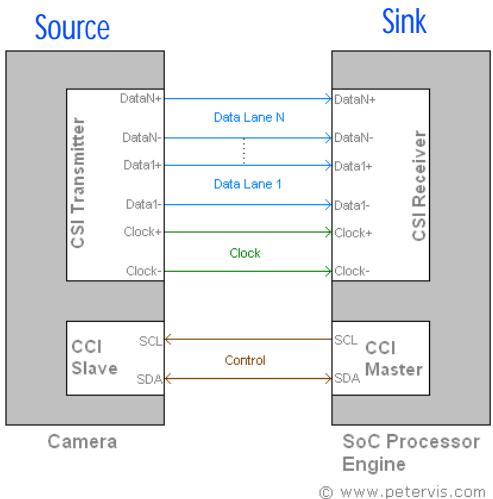


ขา	ชื่อ	หน้าที่
1	Ground	กราวด์
2	Data Lane 1-	ขาลงเลนข้อมูล 1
3	Data Lane 1+	ขาขึ้นเลนข้อมูล 1
4	Ground	กราวด์
5	Clock N	ขาลงคล็อก
6	Clock P	ขาขึ้นคล็อก
7	Ground	กราวด์
8	Data Lane 0-	ขาลงเลนข้อมูล 0
9	Data Lane 0+	ขาขึ้นเลนข้อมูล 0
10	Ground	กราวด์
11		
12		
13	Ground	กราวด์
14	+3.3 V	ไฟเลี้ยงขนาด 3.3 โวลท์
15	+3.3 V	ไฟเลี้ยงขนาด 3.3 โวลท์

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

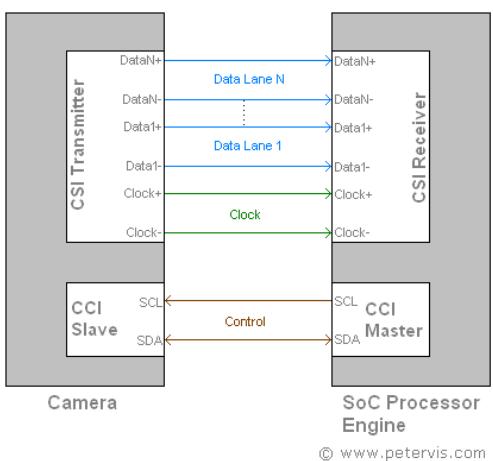
12

6.3 สัญญาณ CSI สำหรับเชื่อมต่อกล้องขนาดเล็ก



- มาตรฐาน CSI: Camera Serial Interface มีความคล้ายคลึงกับสัญญาณ DSI ซึ่งกำหนดโดยองค์กรเดียวกัน คือ MIPI
- ข้อมูลภาพจากกล้องจะส่งผ่านสายด้วยเลนข้อมูลจำนวนหนึ่ง เพื่อไปรวมกันเป็นภาพเดียวที่ปลายทางแต่ละเลนมีการส่งข้อมูลที่ล่าไบท์และอนุกรม
- ข้อมูลจะส่งแบบบีบอัดตามสัญญาณคลือก และสัญญาณควบคุมมาตรฐาน I2C

6.3 สัญญาณ CSI สำหรับเชื่อมต่อกล้องขนาดเล็ก



6.5 สัญญาณภาพและเสียงสำหรับจอทีวี

- บอร์ด Pi3 สามารถเชื่อมต่อกับจอทีวี โดยใช้สัญญาณภาพและเสียง
- สัญญาณภาพ เรียกว่า สัญญาณคอมโพสิทวีดีโอ มีความละเอียดต่ำกว่าสัญญาณ HDMI
- สัญญาณเสียงเป็นแบบสเตอริโอ สำหรับลำโพงซ้ายและขวา
- สัญญาณเสียง มีรายละเอียดในหัวข้อถัดไป

6.5 สัญญาณภาพและเสียงสำหรับจอทีวี



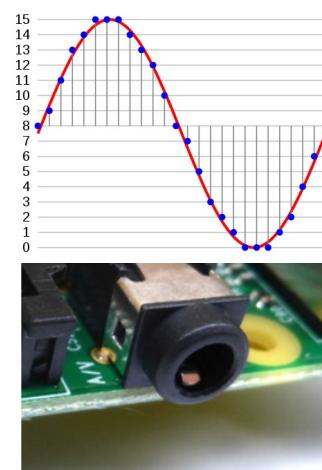
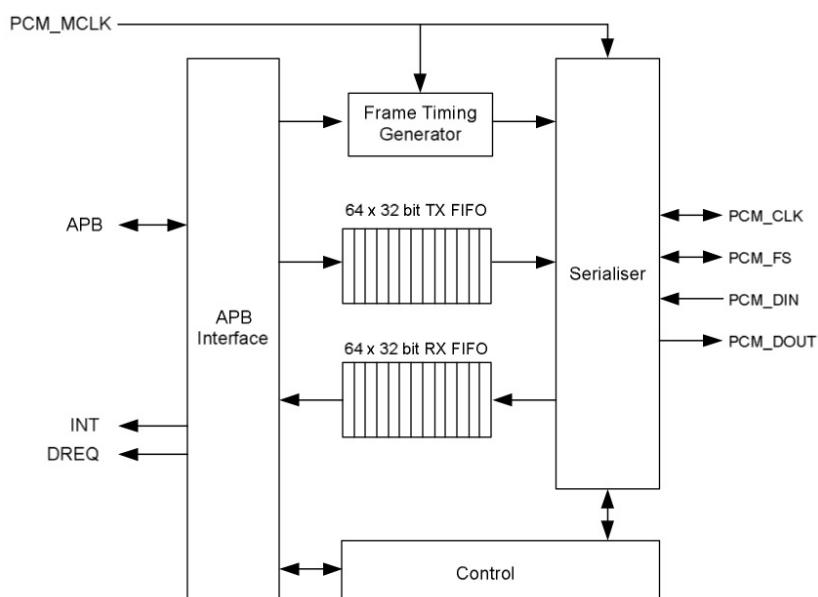
6.4 สัญญาณ PCM สำหรับสัญญาณเสียง

- สัญญาณชนิด PCM คือ สัญญาณดิจิทัลพื้นฐานเกิดจากการแปลงสัญญาณอนาล็อกเป็นดิจิทัล (Analog to Digital: A2D) เป็นเลขจำนวนเต็มชนิดไม่มีเครื่องหมาย
- นิยมแพร่หลายในอุตสาหกรรมปัจจุบัน และใช้กับแผ่นซีดี (Compact Disc) โทรศัพท์บ้านพื้นฐาน และอื่นๆ
- ชิป BCM2837 บนบอร์ด Pi3 สามารถแปลงข้อมูลเสียงที่ได้จากการประมวลผลในรูปแบบ PCM และแปลงสัญญาณดิจิทัลให้เป็นสัญญาณอนาล็อกเพื่อส่งต่อให้กับลำโพงภายนอก

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

17

6.4 สัญญาณ PCM สำหรับสัญญาณเสียง



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

18

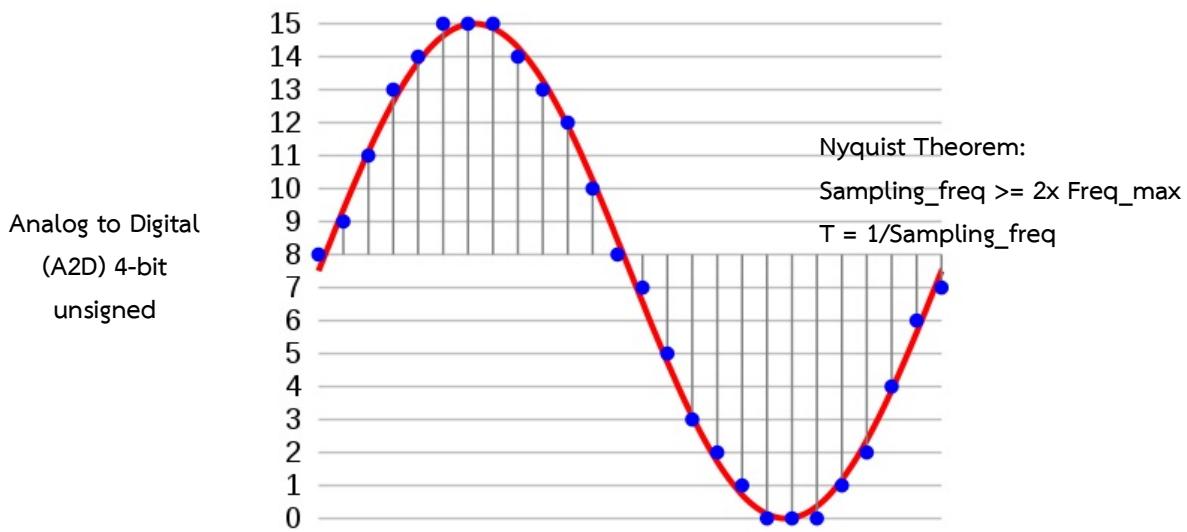
6.4 สัญญาณ PCM สำหรับสัญญาณเสียง

- Nyquist Theorem: $\text{Sampling_freq} \geq 2 \times \text{Freq_max}$, $T = 1/\text{Sampling_freq}$
- รูปคลื่นไอน์ (Sine Wave) และการสุ่มค่าของคลื่นไอน์นี้ ด้วยความถี่สูงเป็น 26 เท่าของความถี่เดิม แล้วทำการแปลงเป็นสัญญาณดิจิทัลชนิด PCM ด้วยความละเอียด 16 ระดับให้กลับเป็นข้อมูลขนาด 4 บิตต่อการสุ่ม 1 ครั้ง
- สัญญาณเสียงสนทนาก่อนโทรศัพท์จะสุ่มด้วยความถี่ 8,000 ครั้งต่อวินาที ซึ่งจะตรงกับคาบเวลา $1/8,000 = 125$ ไมโครวินาที ด้วยความละเอียด 256 ระดับ หรือ 8 บิต
- สัญญาณเสียงเพลงคุณภาพระดับแผ่นเสียงจะสุ่มด้วยความถี่ 44,100 ครั้งต่อวินาที ซึ่งจะตรงกับคาบเวลา $1/44,100 = 22.67$ ไมโครวินาที ด้วยระดับความละเอียด 65,536 ระดับ เพื่อให้เป็นข้อมูลขนาด 16 บิต

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

19

6.4 สัญญาณ PCM สำหรับสัญญาณเสียง



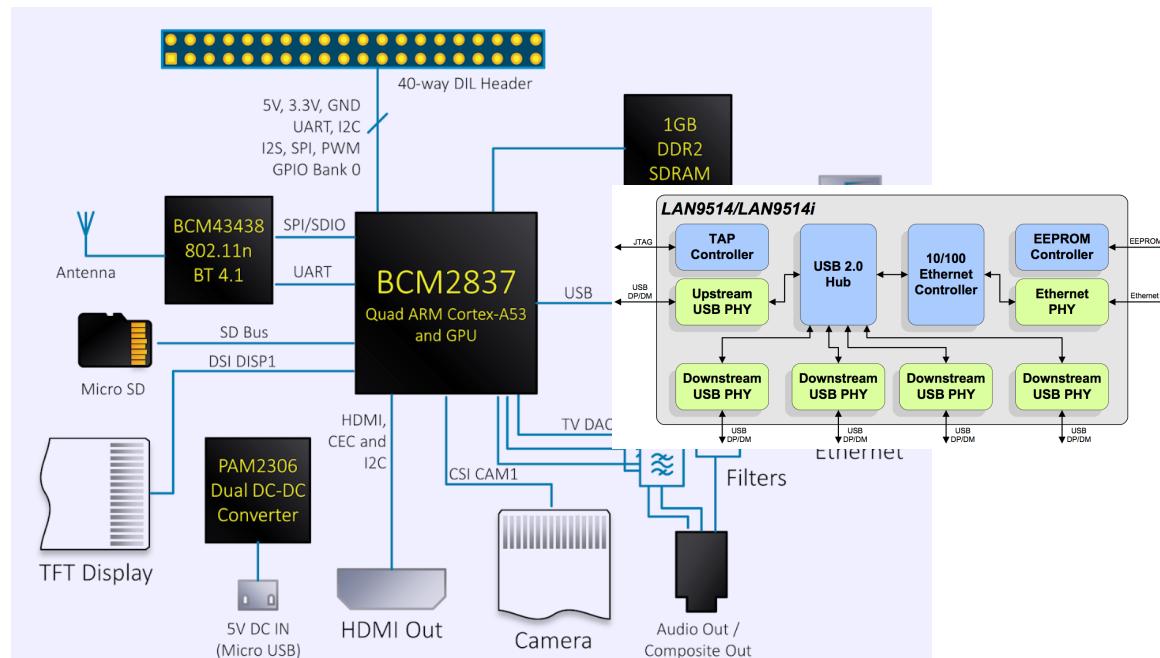
Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

20

6.6 สัญญาณ USB 2.0 สำหรับอุปกรณ์ต่อพ่วงต่างๆ

- ในตำราเล่มนี้จะกล่าวถึง USB เวอร์ชัน 2.0 ซึ่งเป็นพื้นฐานและมีคุณสมบัติ ดังนี้
 - สามารถโอนถ่ายข้อมูลทั่วไป สัญญาณเสียง และสัญญาณภาพได้สูงสุดถึง
 - 1.5 (Low Speed) 12 (Full Speed) และ 48 (High Speed) เมกะบิตต่อวินาที
- สามารถจ่ายไฟเลี้ยงความต่างศักย์ 5 โวลท์ 0.5 แอม培ร์ให้แก่อุปกรณ์ขนาดเล็ก และสูงสุด 1 แอม培ร์สำหรับพอร์ตพิเศษ
- สายเคเบิลมีความยาวไม่เกิน 5 เมตร เนื่องจากความต้านทานของสายจะทำให้เกิดโวลต์เจตกรคร่อม (Voltage Drop) ในสาย จนทำให้ความต่างศักย์ไปเลี้ยงอุปกรณ์ไม่เพียงพอ
- "Hot Swapping" รองรับการต่อเข้า/ออก ตลอดเวลา และรีเซ็ตอุปกรณ์ที่ต่ออยู่โดยไม่ต้องรีเซ็ตหรือรีบูตระบบโอเอส

6.6 สัญญาณ USB 2.0 สำหรับอุปกรณ์ต่อพ่วงต่างๆ



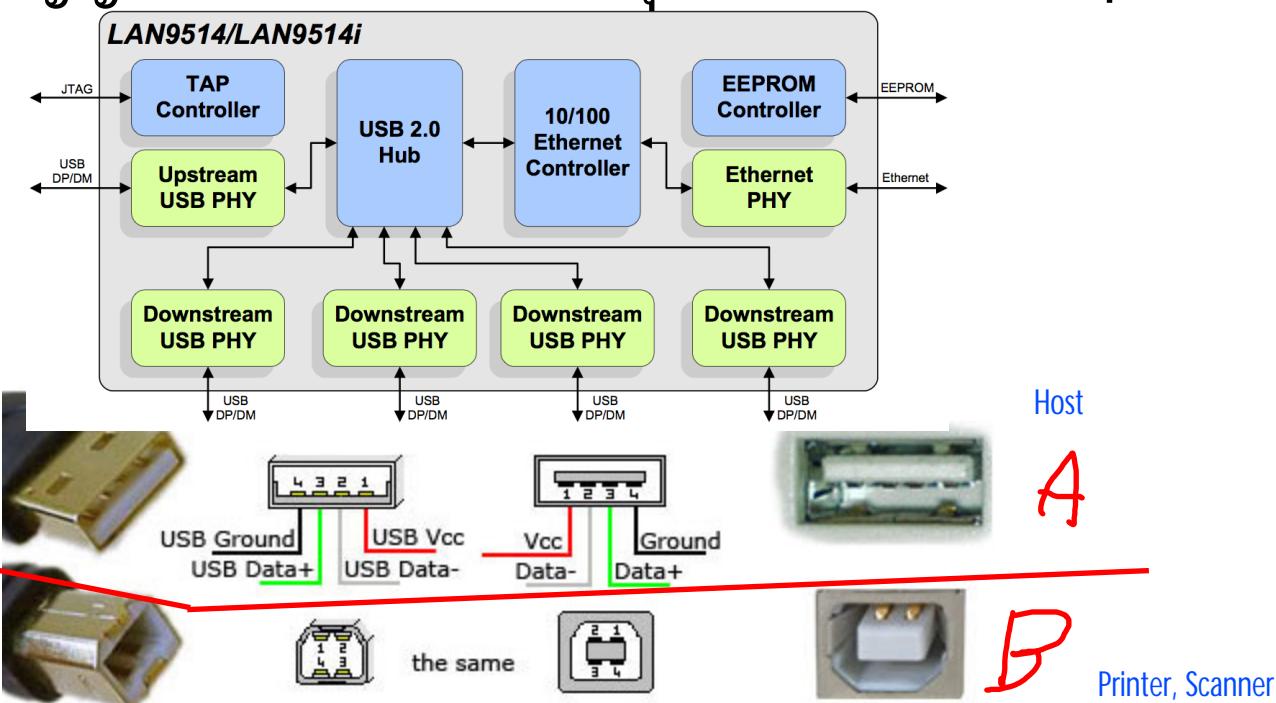
6.6 สัญญาณ USB 2.0 สำหรับอุปกรณ์ต่อพ่วงต่างๆ

- เนื่องจากภายในชิป BCM2837 จะมี Root Hub เพียง 1 พอร์ต
- โครงสร้างของไอซี LAN9514 ถูกออกแบบให้ LAN9514 มี USB Hub (Upstream) จำนวน 1 พอร์ต เพื่อเชื่อมกับ Root Hub ในชิป BCM2837 และขยายจำนวนพอร์ต (Downstream) เพิ่มเป็น 4 พอร์ต เพื่อต่อเข้ามายังบอร์ดแม่ส์ และอุปกรณ์ USB อื่นๆ
- ภายใน LAN9514 ยังมีโมดูล Ethernet สำหรับเชื่อมต่อกับเครือข่ายอินเตอร์เน็ตแบบใช้สาย ซึ่งจะได้กล่าวต่อไป
- ในทางปฏิบัติชิป LAN9514 มีพอร์ต IEEE 1149.1 TAP (Test Access Port) CONTROLLER เพื่อใช้สำหรับทดสอบวงจรภายใน

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

23

6.6 สัญญาณ USB 2.0 สำหรับอุปกรณ์ต่อพ่วงต่างๆ



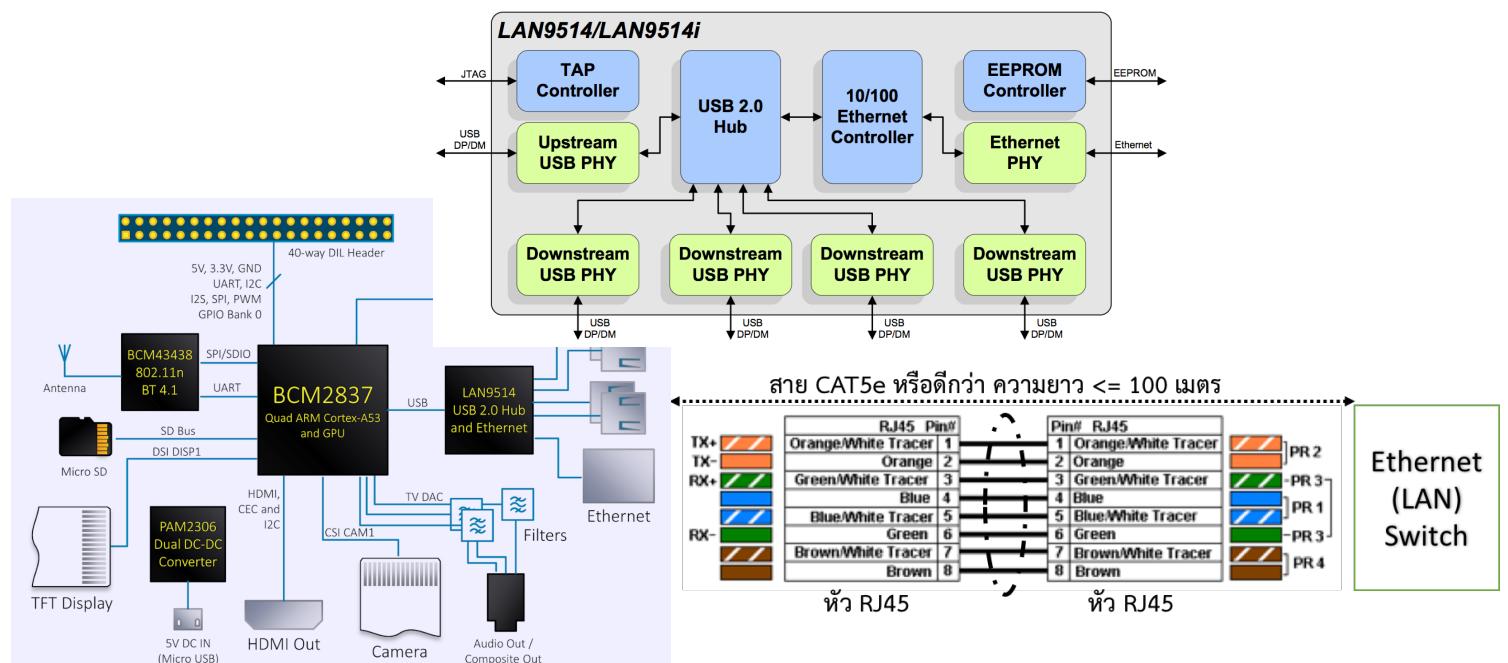
Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

24

6.7 สัญญาณ Ethernet สำหรับสายเชื่อมต่อ กับ อินเตอร์เน็ต

- บอร์ด Pi3 นี้ รองรับการเชื่อมต่อ Ethernet ตามมาตรฐาน IEEE 802.3 ชนิด 10/100 BaseT ซึ่ง เป็นที่นิยมทั้งในอดีตและปัจจุบัน ด้วยอัตรา 10/100 เมกะบิตต่อวินาที (Mbps)
- สายที่ใช้มีชื่อว่าสาย CAT5e หรืออาจจะใช้สายที่มาตรฐานสูงกว่าได้ เช่น CAT6 CAT6A เป็นต้น โดยจะต้องเชื่อมบอร์ดเข้ากับอุปกรณ์เครือข่าย ที่เรียกว่า Ethernet Switch ตามลำดับชั้น เพื่อ สุดท้ายเชื่อมต่อกับเครือข่ายอินเตอร์เน็ต
- เพื่อให้บอร์ดทำหน้าที่เป็น เซิร์ฟเวอร์ (Server) เช่น เว็บเซิร์ฟเวอร์ FTP เซิร์ฟเวอร์ เป็นต้น สำหรับ อุปกรณ์ IoT จากการเชื่อมต่อกับเซ็นเซอร์ต่างๆ

6.7 สัญญาณ Ethernet สำหรับสายเชื่อมต่อ กับ อินเตอร์เน็ต



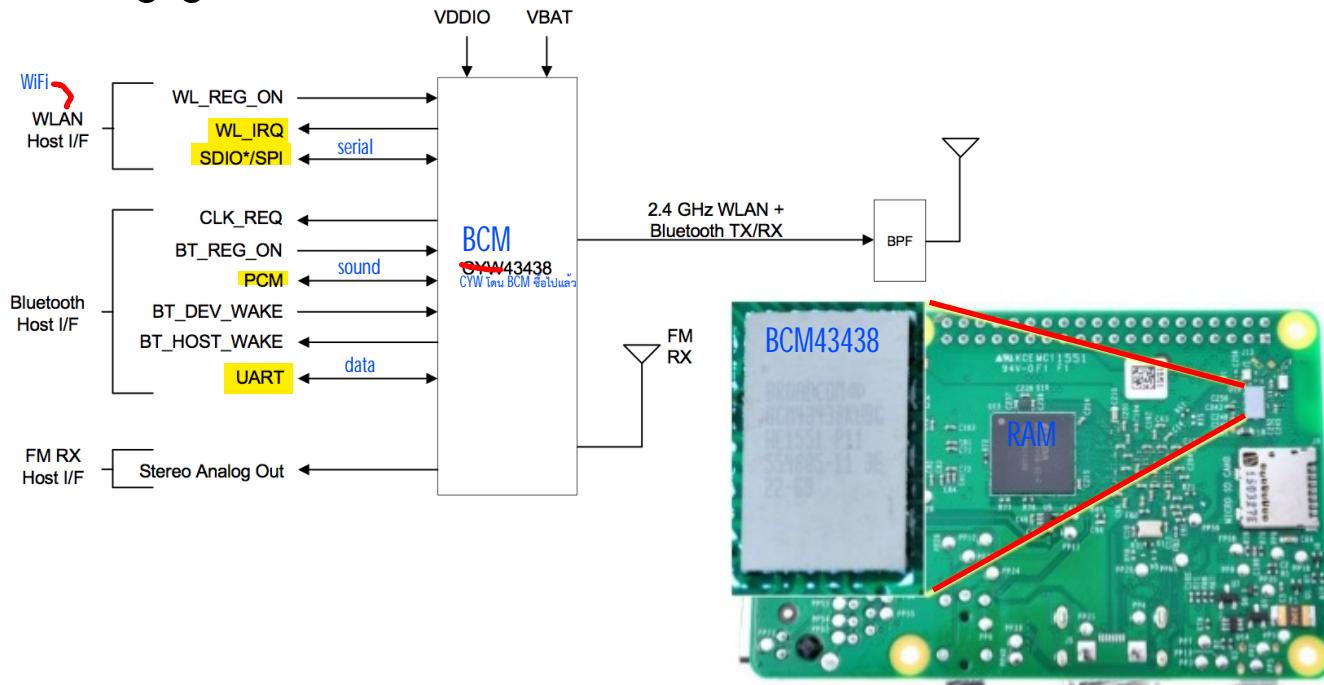
6.8 สัญญาณ WiFi และ Bluetooth สำหรับการสื่อสารไร้สาย

- ชิป BCM 43438 บนบอร์ด RPi3 รองรับสัญญาณ IEEE 802.11b/g/n ที่ย่านความถี่คลื่น파ห์ 2.4 GHz เท่านั้น และสัญญาณ Bluetooth เวอร์ชัน 4.1
- บล็อกไดอะแกรมของชิป BCM 43438 ประกอบด้วยขาสัญญาณเชื่อมต่อเสาอากาศ และขาเชื่อมต่อกับไมโครคอนโทรลเลอร์ (Host Interface)
- หัวส่งสัญญาณใช้สายอากาศ (Antenna) และความถี่พาร์คท์หลักในย่านความถี่ 2.4 กิกะเฮิทซ์ เดียว กัน บลูทูธใช้หลักการ Frequency Hopping และกำลังส่งที่ต่ำกว่าสัญญาณ WiFi ทำให้มีเกิดการรบกวนกัน

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

27

6.8 สัญญาณ WiFi และ Bluetooth สำหรับการสื่อสารไร้สาย

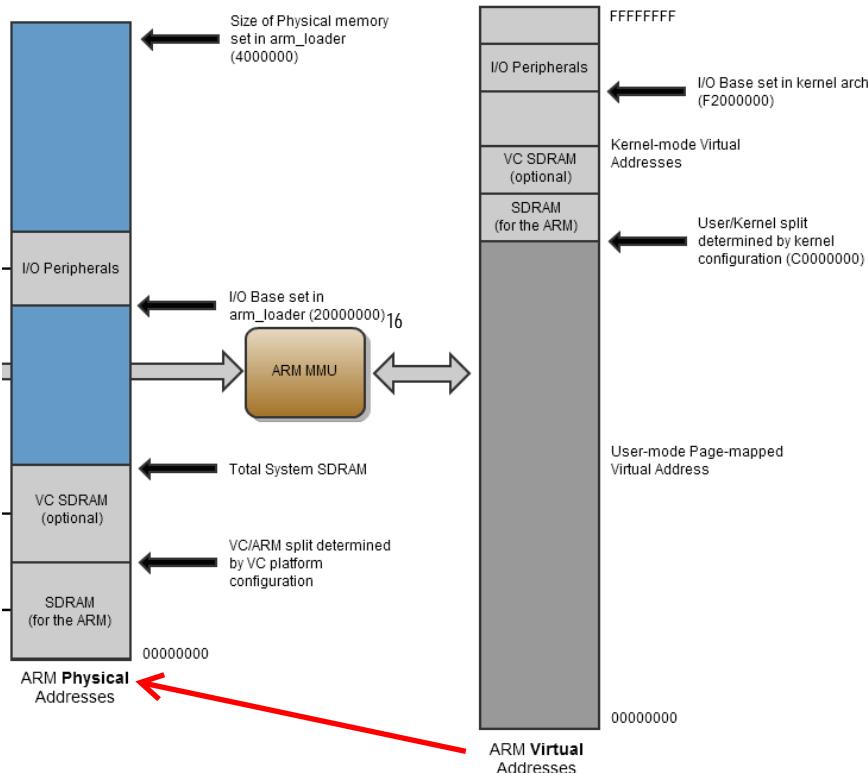


Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

28

6.9 หลักการ Memory Mapped Input/Output

ภายในชิป BCM283x มีวงจรหรืออุปกรณ์/อินพุทเอาท์พุท (I/O Peripherals) มีจำนวนมาก การเข้ามาร่วมกับอุปกรณ์เหล่านี้จะใช้การตั้งค่าแอดเดรสบัส (Bus Address) เพื่ออ่านข้อมูลและเขียนข้อมูลเหล่านี้แล้วmap (Map) แอดเดรสบัสเหล่านี้ไปบนแอดเดรสหน่วยความจำภายในภาพ



6.9 หลักการ Memory Mapped Input/Output

การอ่านหรือเขียนข้อมูลไปยังแอดเดรสภายในภาพเหล่านี้ ทำได้การใช้คำสั่ง LDR และ STR เมื่องานกับหน่วยความจำปกติที่ว่าไป โดยผู้ผลิตหารดไวร์ได้กำหนดหมายเลขและแอดเดรสของบัสตามตารางที่ 6.4 โดยแอดเดรสบัสเริ่มต้นที่หมายเลข 0x7E000000 หรือ 0x7E00_0000

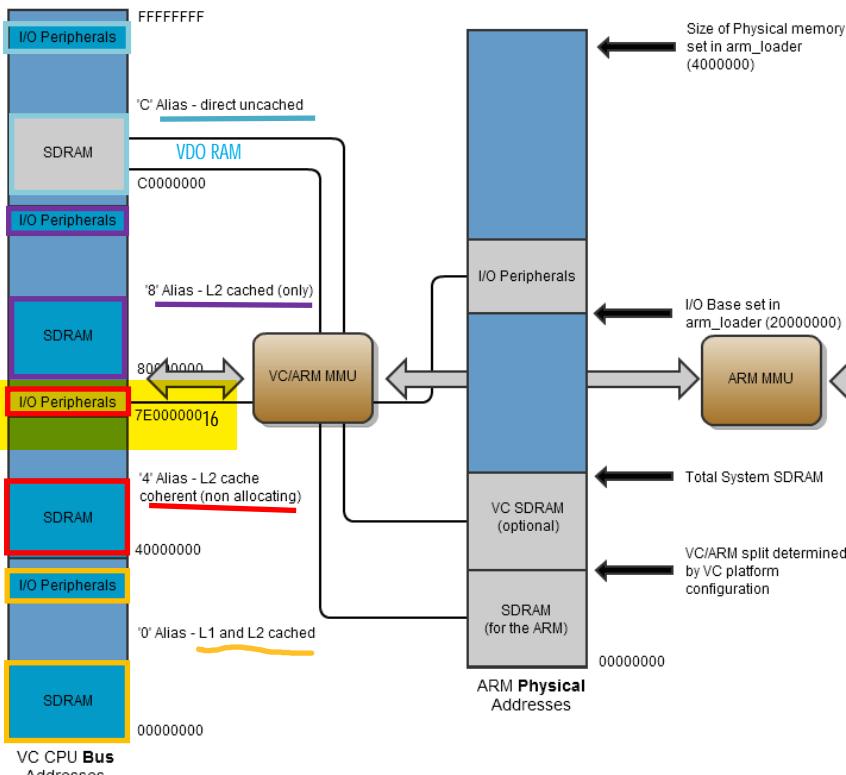
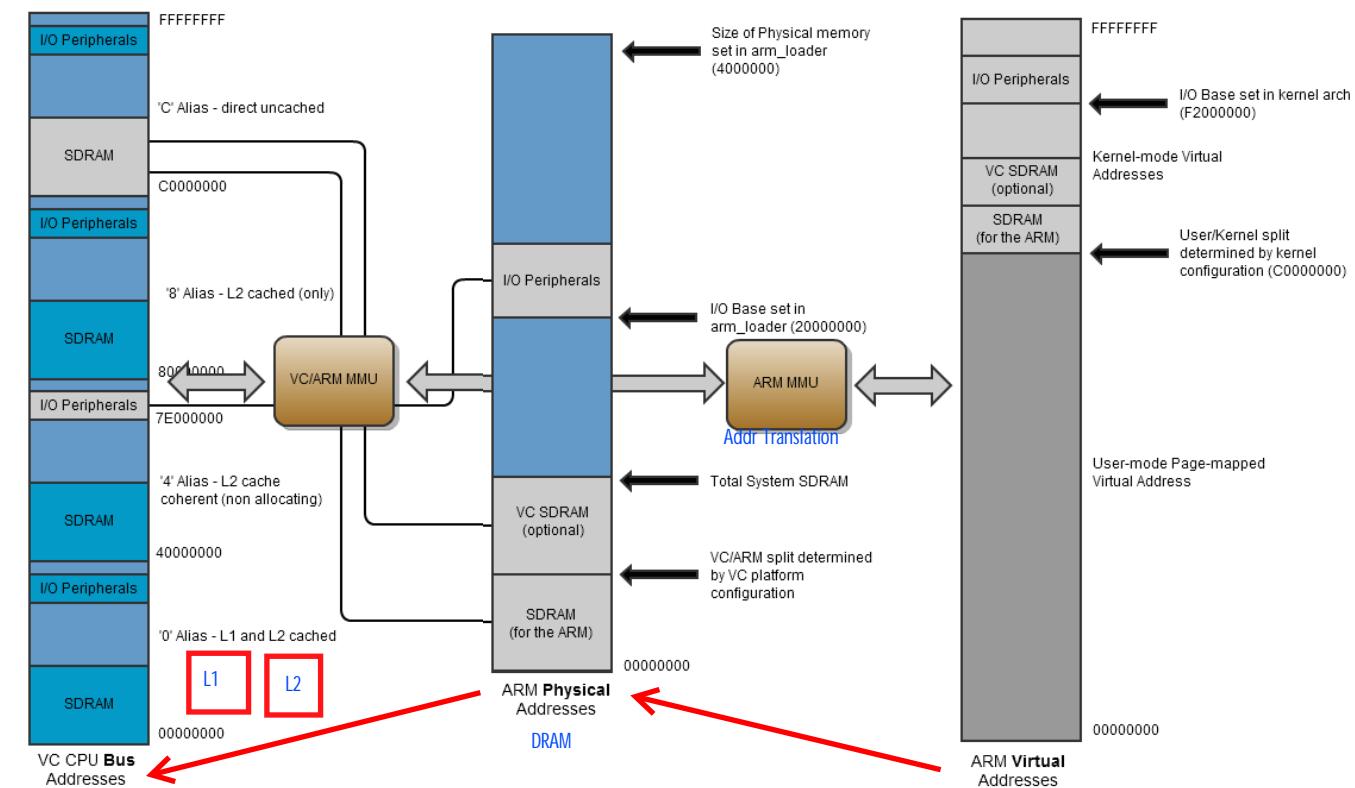
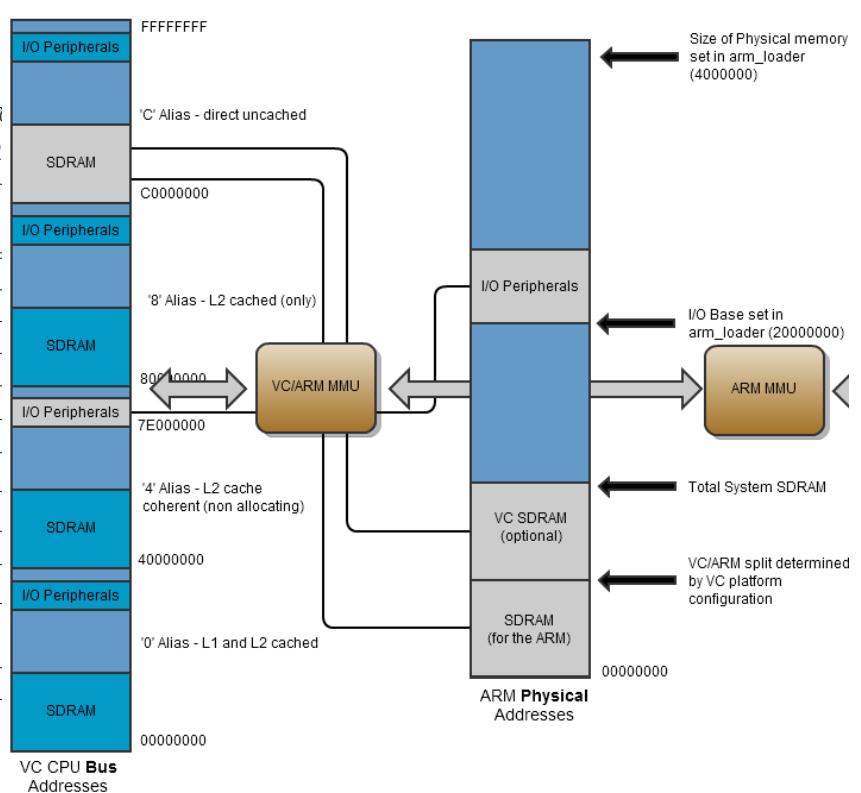


Table 6.4: ตารางแสดงโครงสร้างเมมทั้งหมดที่หมายเลขอาร์ม (ARM Peripherals) และหัวข้อที่มา: Broadcom (2012)

แอดเดรสบัส (Bus Address)	ชื่อ (Name)
0x7E00_0000	...
0x7E00_1000	...
0x7E00_2000	...
0x7E00_3000	System Timer
0x7E00_7000	DMA Controller
0x7E00_B000	Interrupt Register
0x7E00_B400	Timer
0x7E20_0000	General Purpose I/O
0x7E20_3000	Pulse Code Modulation
0x7E21_5000	mini UART, SPI1, SPI2
0x7E30_0000	External Mass Media Controller
0x7E98_0000	Universal Serial Bus

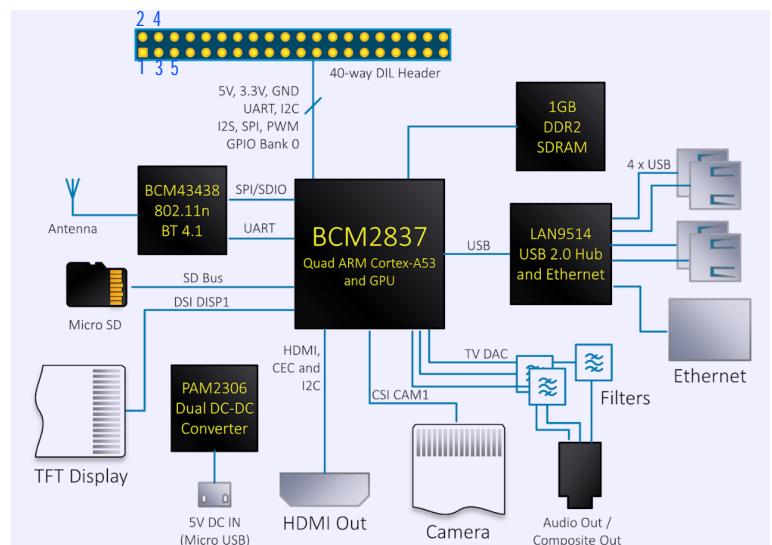


6.10 หัวเชื่อมต่อ 40 ขา (40-Pin Header)

Raspberry Pi 3 GPIO Header				
Pin#	NAME	ขา 1 จะเป็นขาเสี่ยง	NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)		DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Rev. 2
29/02/2016

www.element14.com/RaspberryPi



6.10 หัวเชื่อมต่อ 40 ขา (40-Pin Header)

Raspberry Pi 3 GPIO Header			
Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Rev. 2
29/02/2016

www.element14.com/RaspberryPi

ขา	ชื่อ	ตัวเลือก0	ตัวเลือก1	ตัวเลือก2	ตัวเลือก3	ตัวเลือก4
1	3.3V					
2	5.0V					
3	GPIO02	SDA1	SA3	LCD_VSYNC		
4	5.0V					
5	GPIO03	SCL1	SA2	LCD_HSYNC		
6	GND					
7	GPIO04	GPCLK0	SA1	DPI_D0		
8		TxD0	SD6	DPI_D10		
9	GND					
10	GPIO15	RxD0	SD7	DPI_D11		
11	GPIO17	FL1	SD9	DPI_D13	RTS0	SPI1_CE1_N
12		PCM_CLK	SD10	DPI_D14		SPI1_CE0_N
13	GPIO18	SD0_DAT3	TE1	DPI_D23	SD1_DAT3	ARM_TMS
14	GND					
15	GPIO22	SD0_CLK	SD14	DPI_D18	SD1_CLK	ARM_TRST
16	GPIO23	SD0_CMD	SD15	DPI_D19	SD1_CMD	ARM_RTCK
17	3.3V					
18	GPIO24	SD0_DAT0	SD16	DPI_D20	SD0_DAT0	ARM_TDO
19	GPIO10	SD0_MOSI	SD2	DPI_D6		
20	GND					

6.11 ขา GPIO (General Purpose Input Output)

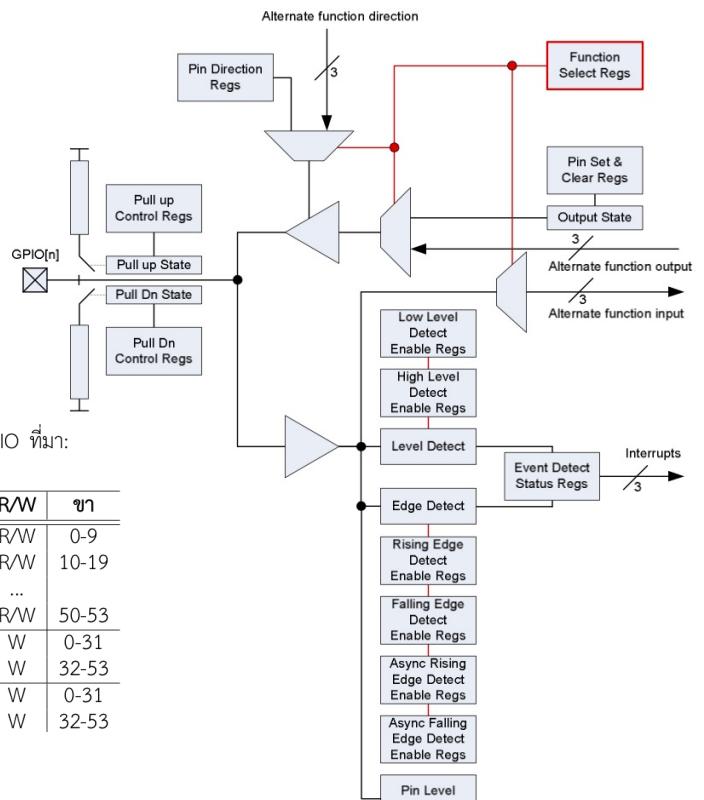


Table 6.6: ตารางแสดงผลเดรสนิหน่วยความจำเริ่มต้นที่หมายเลข 0x7E20_0000 สำหรับ GPIO ที่มา: Broadcom (2012)

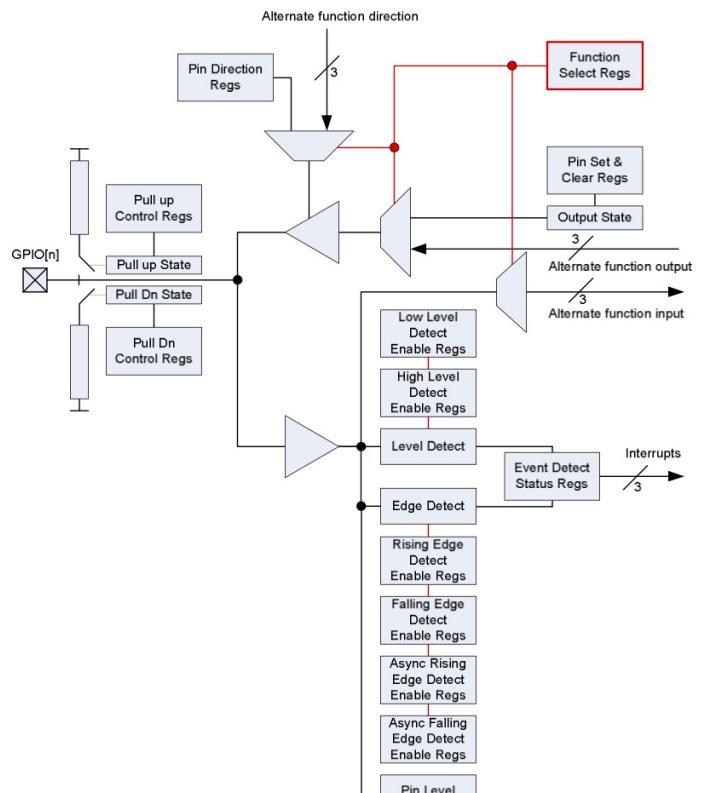
แอดเดรสบัส	รีจิสเตอร์	รายละเอียด	จำนวนบิต	R/W	ขา
0x7E20_0000	GPFSEL0	GPIO Function Select 0	32	R/W	0-9
0x7E20_0004	GPFSEL1	GPIO Function Select 1	32	R/W	10-19
...
0x7E20_0014	GPFSEL5	GPIO Function Select 5	12	R/W	50-53
0x7E20_001C	GPSET0	GPIO Pin Output Set 0	32	W	0-31
0x7E20_0020	GPSET1	GPIO Pin Output Set 1	22	W	32-53
0x7E20_0028	GPCLR0	GPIO Pin Output Clear 0	32	W	0-31
0x7E20_002C	GPCLR1	GPIO Pin Output Clear 1	22	W	32-53

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

35

6.11 ขา GPIO (General Purpose Input Output): Function Select Register

- 000 = ขาเข้า (Input)
- 001 = ขาออก (Output)
- 100 = ตัวเลือก 0 ในตารางที่ 6.5
- 101 = ตัวเลือก 1 ในตารางที่ 6.5
- 110 = ตัวเลือก 2 ในตารางที่ 6.5
- 111 = ตัวเลือก 3 ในตารางที่ 6.5
- 011 = ตัวเลือก 4 ในตารางที่ 6.5
- 010 = ตัวเลือก 5 ในตารางที่ 6.5



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

36

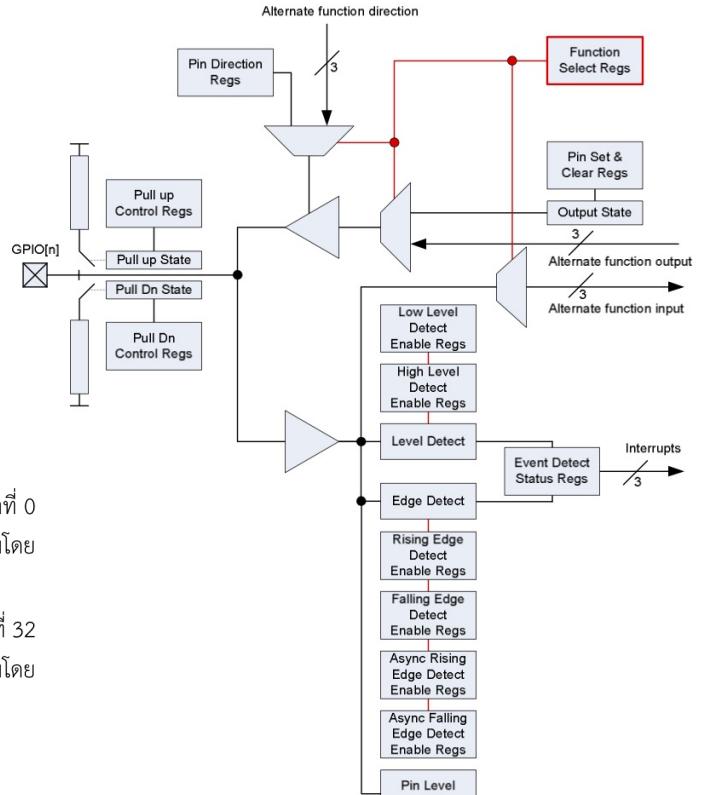
6.11 ขา GPIO

(General Purpose Input Output):

Register Pin Set & Pin Clear

- การตั้งขาเป็นขาเอาท์พุท (000)

- รีจิสเตอร์ GPSET0 (GPIO Pin Output Set) ใช้สำหรับตั้งค่าเอาท์พุทเป็นโลจิก 1
 - * แอดเดรส 0x7E20_001C - 0x7E20_001F จำนวน 4 บิต หรือ 32 บิต ควบคุมขาที่ 0 - 31 จำนวน 32 ขา ละ 1 บิต ขาที่ 0 ควบคุมโดยบิตที่ 0 จนถึงขาที่ 31 ควบคุมโดยบิตที่ 31 ตามลำดับ
 - * แอดเดรส 0x7E20_0020 - 0x7E20_0022 จำนวน 3 บิต หรือ 24 บิต ควบคุมขาที่ 32 - 53 จำนวน 22 ขา ละ 1 บิต ขาที่ 32 ควบคุมโดยบิตที่ 0 จนถึงขาที่ 53 ควบคุมโดยบิตที่ 21 ตามลำดับ
- รีจิสเตอร์ GPCLR0 (GPIO Pin Output Clear) ใช้สำหรับตั้งค่าเอาท์พุทเป็นโลจิก 0



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

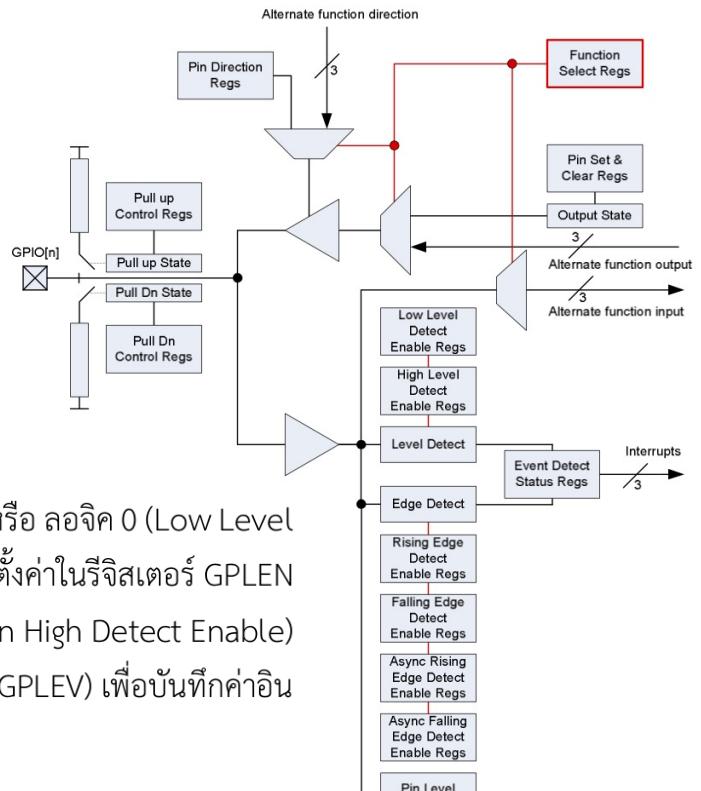
37

6.11 ขา GPIO

(General Purpose Input Output):

I/P Level Detect

- การตรวจจับระดับสัญญาณ (Level Detect) แบ่งเป็น สัญญาณต่ำ หรือ โลจิก 0 (Low Level Detect) สัญญาณสูง หรือ โลจิก 1 (High Level Detect) โดยการตั้งค่าในรีจิสเตอร์ GPLEN (GPIO Pin Low Detect Enable) และรีจิสเตอร์ GPHEN (GPIO Pin High Detect Enable) ตัวไดตัวหนึ่งให้เท่ากับโลจิก 1 ที่เหลือเป็น 0 รีจิสเตอร์ Pin Level (GPLEV) เพื่อบันทึกค่าอินพุทที่รับเข้ามา



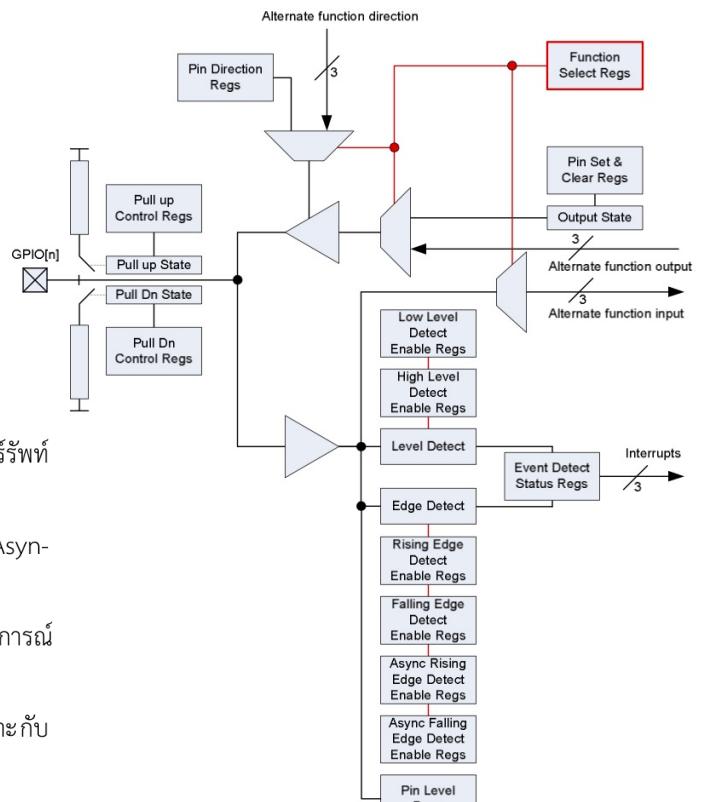
Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

38

6.11 ขา GPIO

(General Purpose Input Output):

I/P Edge Detect



- การตรวจจับขอบสัญญาณ (Edge Detection) แบ่งเป็น

- * ขอบขาขึ้น (Rising) และขอบขาลง (Falling) เมื่อตรวจจับได้ จะส่งสัญญาณอินเทอร์รัพท์ เข้าสู่ซีพียูต่อไป
- * การตรวจจับสัญญาณแบ่งเป็น ชนิดซิงโครนัส (Synchronous) และ อะซิงโครนัส (Asynchronous) ความหมายคือ ซิงโครไนซ์ (Synchronize) กับสัญญาณคล็อกหรือไม่
 - ชนิดซิงโครนัส การตรวจจับจะขึ้นกับความถี่ของสัญญาณคล็อก เหมาะกับเหตุการณ์ที่นานๆ แค่เดียว
 - ชนิดอะซิงโครนัส การตรวจจับจะไม่ขึ้นกับความถี่ของสัญญาณคล็อก เหมาะกับเหตุการณ์ที่นานๆ แค่เดียว

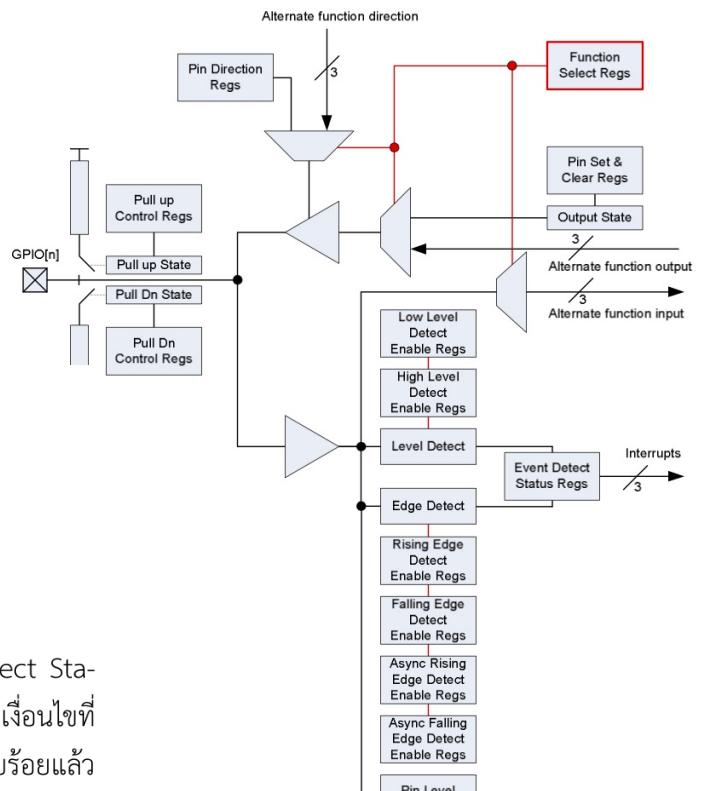
Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

39

6.11 ขา GPIO

(General Purpose Input Output):

I/P Edge Detect



โดยการควบคุมที่รีจิสเตอร์ทั้งสี่นี่ ได้แก่

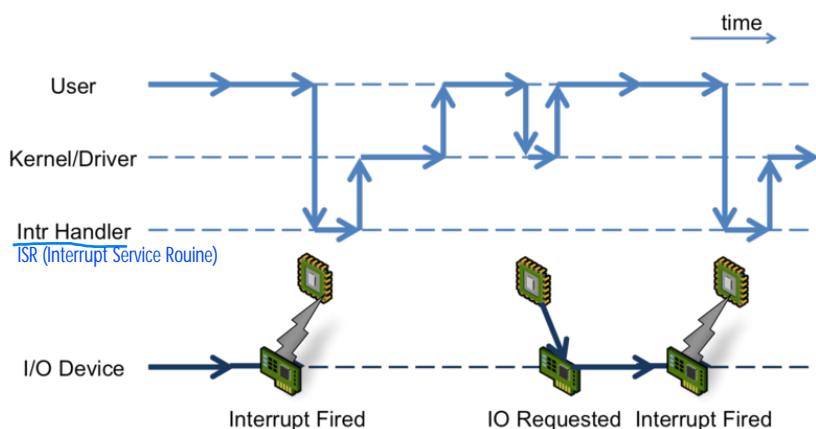
- รีจิสเตอร์ GPREN (GPIO Synchronous Rising Edge Enable) และ
- รีจิสเตอร์ GPFEN (GPIO Synchronous Falling Edge Enable)
- รีจิสเตอร์ GPAREN (GPIO Asynchronous Rising Edge Enable) และ
- รีจิสเตอร์ GPAFEN (GPIO Asynchronous Falling Edge Enable)

เกิดเป็นสัญญาณอินเทอร์รัพท์รองขอไปยังซีพียู รีจิสเตอร์ Pin Event Detect Status (GPEDS) จะเก็บค่าล็อกิค 1 เมื่อเมื่อตรวจจับเหตุการณ์ที่รับเข้ามาได้ตามเงื่อนไขที่ต้องการ และจะถูกเปลี่ยนเป็นล็อกิค 0 เมื่อซีพียูตอบสนองต่อการร้องขอเรียบร้อยแล้ว

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

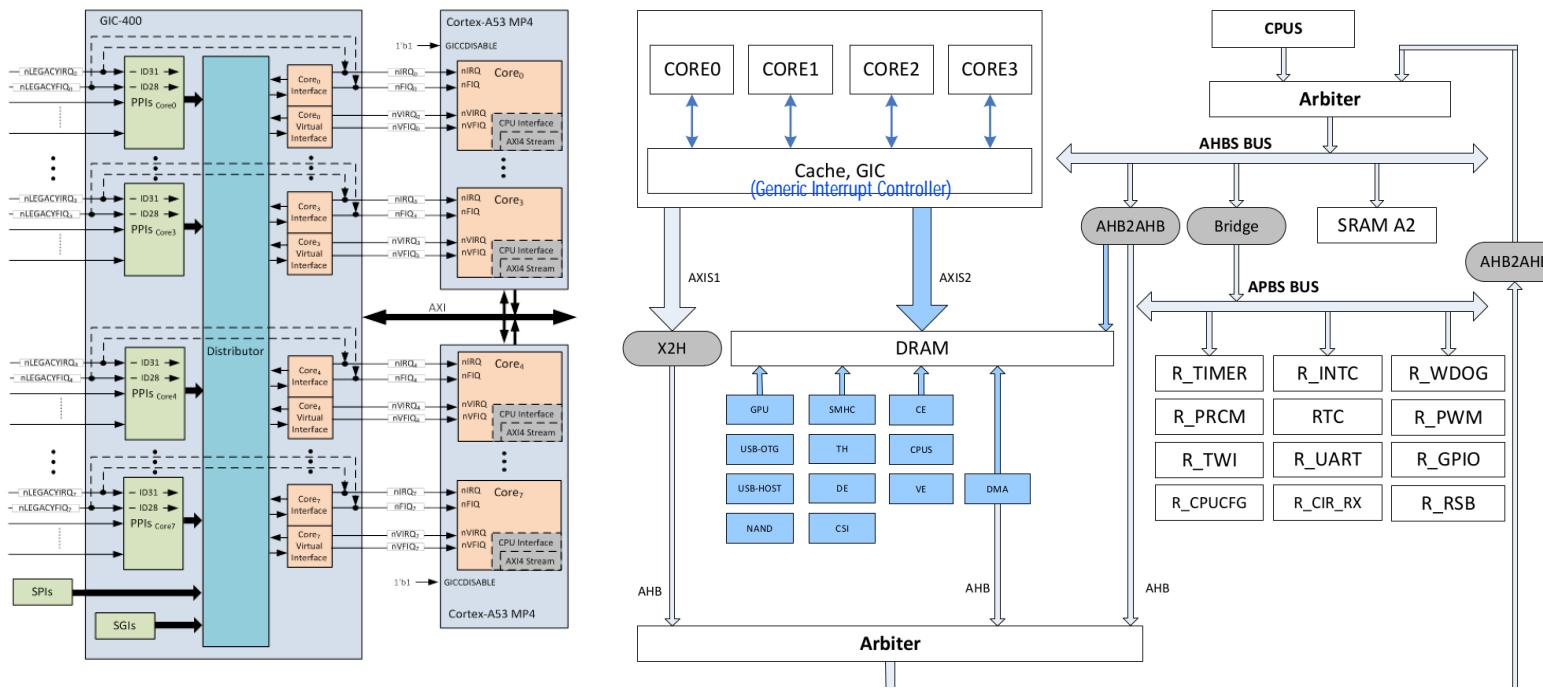
40

6.12 การขัดจังหวะ (Interrupt)



- กลไกการขัดจังหวะเกิดขึ้นจากการใช้งานของร่องรอยที่อยู่ในหน่วยประมวลผล
- Kernel/Driver จะเป็นผู้จัดการข้อมูลการรับส่งข้อมูลแทนโปรแกรมหรือ User
- โปรแกรมย่อยที่ซีพียูจะต้อง Execute สำหรับสัญญาณ Interrupt แต่ละแบบเรียกว่า Interrupt Handler หรือ Interrupt Service Routine (ISR)
- ซีพียูจะไม่เสียเวลาอุดຍุทธการทำงานของร่องรอย IO หากซีพียูไม่มีโอกาสทำงานโปรแกรมอื่นๆ

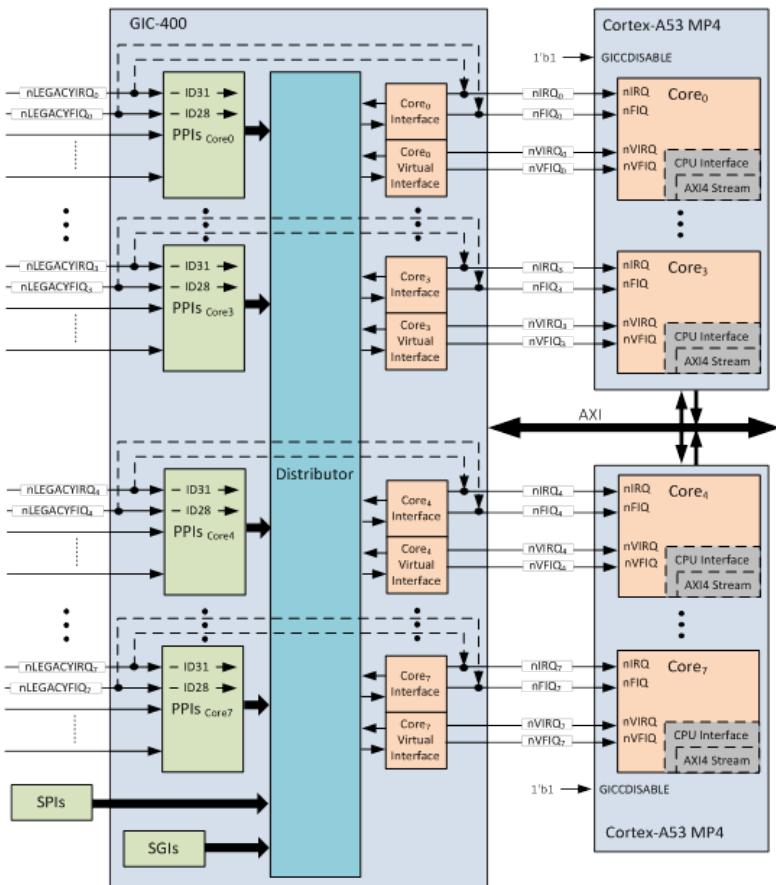
6.12 การขัดจังหวะ (Interrupt)



6.12 การขัดจังหวะ (Interrupt)

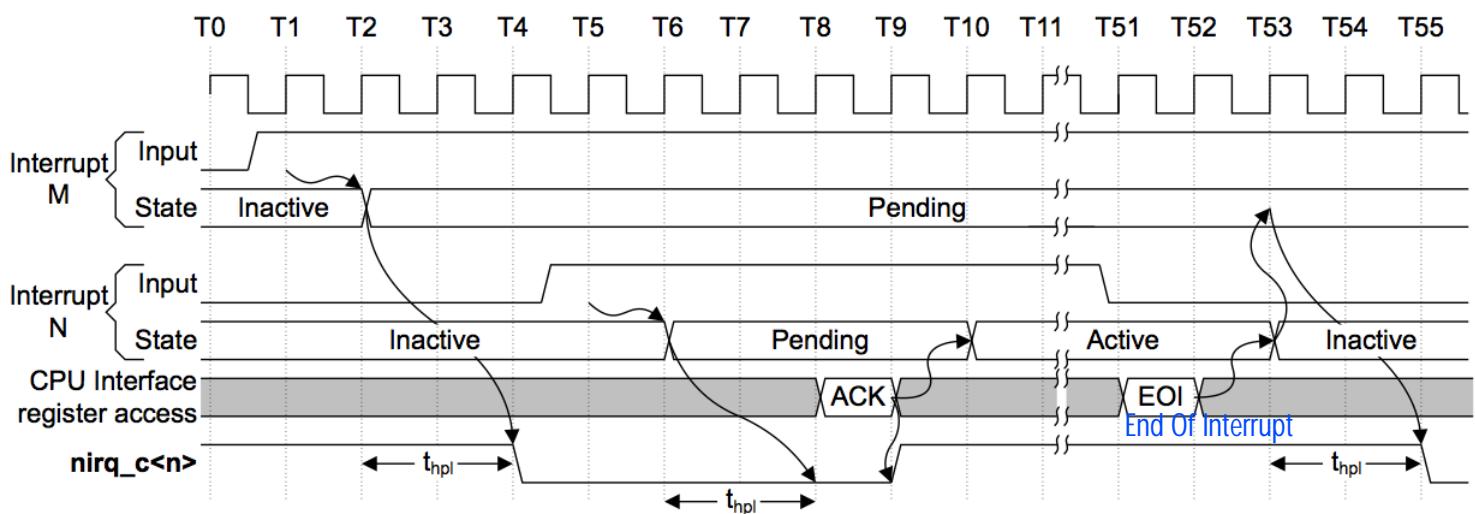
การขัดจังหวะสำหรับ ARM Cortex A รุ่นต่างๆ อาศัยวงจรควบคุม เเรียกว่า Generic Interrupt Controller (GIC) เพื่อรับการขัดจังหวะที่ซับซ้อน เนื่องจากชิปมีจำนวน ARM Cortex A ตั้งแต่ 2 คอร์ขึ้นไป และอุปกรณ์ที่หลากหลาย โดยทั้งหมดเชื่อมต่อกันด้วย ARM Advanced eXtensible Interface (AXI) ซึ่งเป็นบสชนิดหนึ่งมีโครงสร้างและโปรโตคอลที่ซับซ้อน

43



Priority N > M

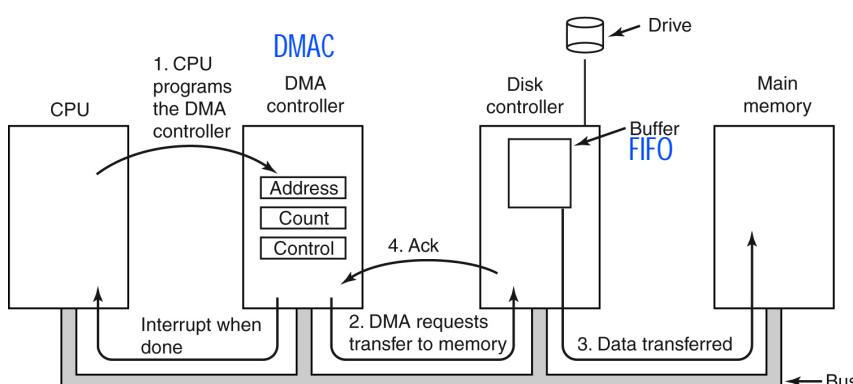
6.12 การขัดจังหวะ (Interrupt)



6.12 การขัดจังหวะ (Interrupt)

- คาบเวลา T1 วงจร Distributor ตรวจจับการร้องขอของ Interrupt M
- คาบเวลา T2 วงจร Distributor พักรการร้องขอจาก Interrupt M
- คาบเวลา T4 วงจรเริ่มต้นกับ ชีพียูตั้งค่า Interrupt M กับสัญญาณร้องขอการขัดจังหวะ $nirq_c < M >$
- คาบเวลา T5 วงจร Distributor ตรวจจับการร้องขอของ Interrupt N ซึ่งมีความสำคัญสูงกว่า Interrupt M จึงทำให้ต้องหยุดการทำงานของ Interrupt M ในคาบเวลาถัดไป
- คาบเวลา T6 วงจร Distributor พักรการร้องขอจาก Interrupt N วงจรเริ่มต้นกับชีพียูตั้งค่า INTID ของ Interrupt N กับสัญญาณร้องขอการขัดจังหวะ $nirq_c < n >$ แทน
- คาบเวลา T8 ชีพียูตอบสนองต่อสัญญาณร้องขอ ด้วยการส่งสัญญาณ ACK
- คาบเวลา T9 ชีพียูกำกับการเปลี่ยนค่า $nirq_c < n >$ โดยเปลี่ยนค่าให้เป็น 1
- คาบเวลา T10 วงจร Distributor ตั้งค่าสถานะของ Interrupt N เป็น Active และเปลี่ยนแปลงค่า Active Status Register
- คาบเวลา T11-T50 ชีพียูตอบสนองโดยการทำงานที่ Interrupt Service Routine ของ Interrupt N หลังจากนั้นอุปกรณ์ยกเลิกการร้องขอ
- คาบเวลา T51 ชีพียูบันทึกค่ารีเซ็ตออร์ End of Interrupt (EOI) ด้วยหมายเลข INTID ของ Interrupt N
- คาบเวลา T52 วงจร Distributor ตั้งค่าสถานะของ Interrupt N เป็น Inactive และเปลี่ยนแปลงค่า Active Status Register.
- คาบเวลา T53 วงจร Distributor แจ้งวงจรเริ่มต้นว่า interrupt M ยังคงอยู่และมีความสำคัญสูงที่สุด ณ เวลานี้
- คาบเวลา T55 วงจรเริ่มต้น ชีพียุส่งสัญญาณร้องขอ $nirq_c < n >$ ให้กลับเป็น 0 เพื่อขัดจังหวะการทำงานของ ชีพียู เป็นลำดับถัดไป

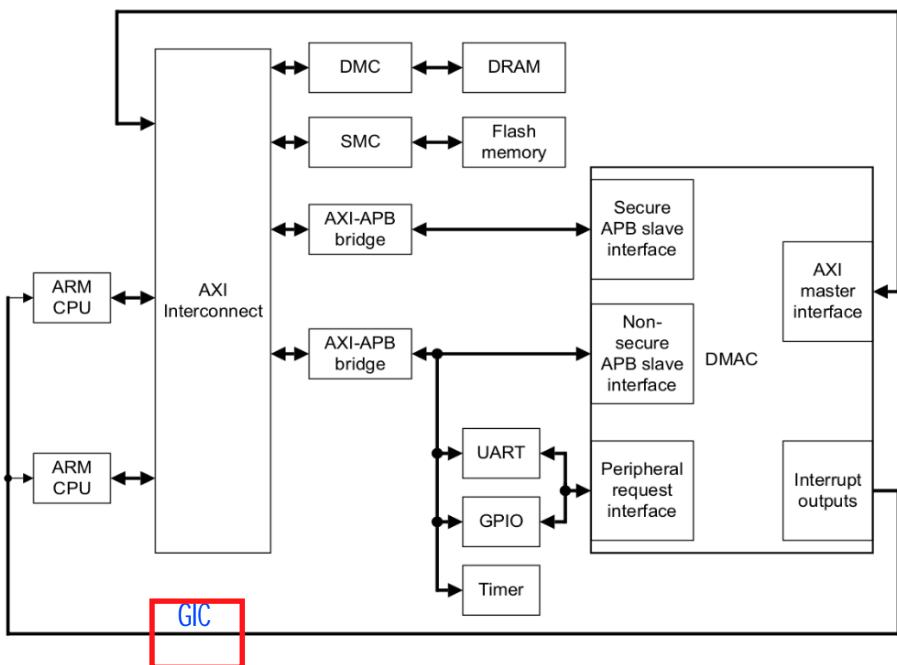
6.13 การเข้าถึงหน่วยความจำโดยตรง (DMA)



1. โปรแกรมสั่งให้ชีพียูตั้งค่าควบคุมกับวงจร DMA Controller
2. DMAC เริ่มต้นโดยการส่งสัญญาณ Request ไปยัง Disk Controller เพื่ออ่านข้อมูลจากดิสก์
3. อ่านข้อมูลจากดิสก์ไปเขียนลงในหน่วยความจำ DRAM จนแล้วเสร็จ
4. Disk Controller ส่งสัญญาณตอบรับ (Acknowledge) ไปยัง DMAC
5. DMAC ส่งสัญญาณ Interrupt ไปยังชีพียู เพื่อแจ้งว่าแล้วเสร็จ

6.13 การเข้าถึงหน่วยความจำโดยตรง

(Direct Memory Access)



การรับข้อมูลจากอุปกรณ์อินพุท DMA จะทำหน้าที่เคลื่อนย้ายข้อมูลจาก FIFO ของวงจรเชื่อมต่อไปยังหน่วยความจำหลัก และการส่งข้อมูลไปยังอุปกรณ์เอาท์พุท ต่างๆ DMA (DMA Controller) จะทำหน้าที่เคลื่อนย้ายข้อมูลจากหน่วยความจำหลัก ไปยัง FIFO ของวงจรนั้นๆ

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

47

6.13 การเข้าถึงหน่วยความจำโดยตรง (DMA)

Table 6.4: ตารางแอดเดรสบัสเริ่มต้นที่หมายเลข 0x7E00_0000 สำหรับอุปกรณ์อินพุทเอาท์พุท (IO Peripherals) และหัวขอ ที่มา: [Broadcom \(2012\)](#)

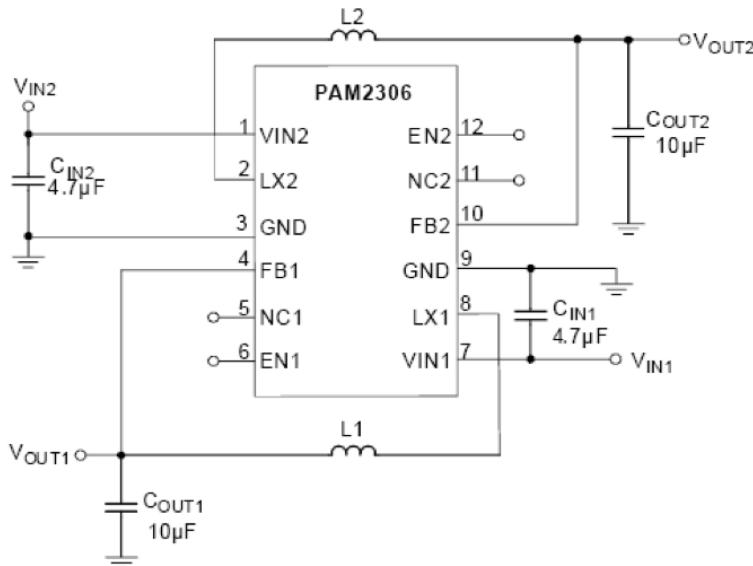
แอดเดรสบัส (Bus Address)	ชื่อ (Name)
0x7E00_0000	...
0x7E00_1000	...
0x7E00_2000	...
0x7E00_3000	System Timer
0x7E00_7000	DMA Controller
0x7E00_B000	Interrupt Register
0x7E00_B400	Timer
0x7E20_0000	General Purpose I/O
0x7E20_3000	Pulse Code Modulation
0x7E21_5000	mini UART, SPI1, SPI2
0x7E30_0000	External Mass Media Controller
0x7E98_0000	Universal Serial Bus

- ขบวนการ DMA เริ่มต้นจาก DMA Controller อ่านค่าการติดตั้งใน CB (Control Block) ในหน่วยความจำไปตั้งค่ารีจิสเตอร์ CONBLK_AD ภายใน DMA ช่อง (Channel) ที่ยังว่างอยู่ หลังจากนั้น วงจร DMA ช่องนั้นจะเริ่มทำงานด้วยตนเอง
- DMA ช่อง 0 เริ่มต้นการตั้งค่าที่แอดเดรส 0x7E00_7000
- DMA ช่อง 1 เริ่มต้นการตั้งค่าที่แอดเดรส 0x7E00_7100
- DMA ช่อง 2 เริ่มต้นการตั้งค่าที่แอดเดรส 0x7E00_7200
- ...
- DMA ช่อง 14 เริ่มต้นการตั้งค่าที่แอดเดรส 0x7E00_7e00
- DMA ช่อง 15 เริ่มต้นการตั้งค่าที่แอดเดรส 0x7EE0_5000

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

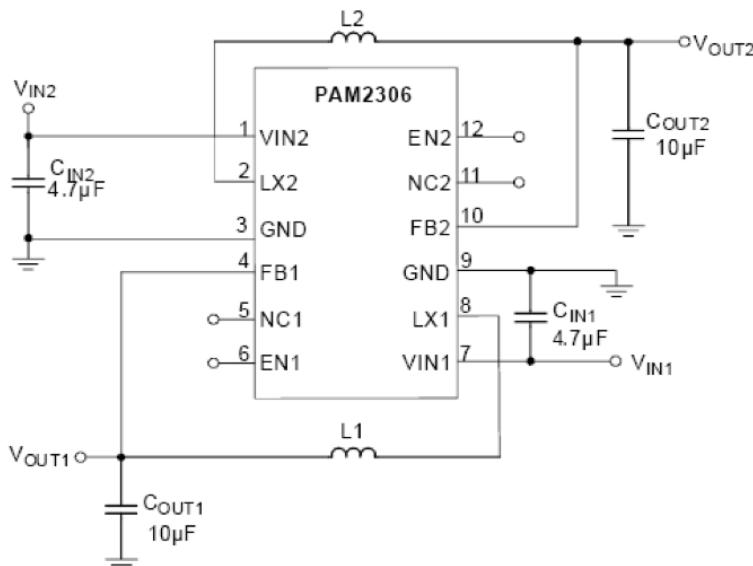
48

6.14 แหล่งจ่ายไฟ (Power Supply) ของบอร์ด Pi3



แหล่งจ่ายไฟของบอร์ดมาจากการแเดปเตอร์ (Adaptor) แปลงไฟฟ้ากระแสสลับ 220 โวลท์เป็นไฟกระแสตรงความต่างศักย์ 5 โวลท์จะให้ผลผ่านพิวส์และไดโอด เพื่อจ่ายไฟให้กับบอร์ด Pi3 และวิจัยถูกแปลงจาก 5 โวลท์ให้ลดลง (Step Down) เป็น 3.3 และ 1.8 โวลท์ด้วยชิป PAM2306 และ 1.2 โวลท์ด้วยชิป RT8088A ตามลำดับ

6.14 แหล่งจ่ายไฟ (Power Supply) ของบอร์ด Pi3



V_{OUT} (โวลท์)	L (μH)
1.2	2.2
1.5	2.2
1.8	2.2
2.5	4.7
3.3	4.7

สรุปห้ายบท

(ของ ARM)

- ซีพีyu หน่วยความจำหลัก และวงจร (ไมค์卢) ด้านอินพุท/เอาท์พุท และอุปกรณ์สำรองข้อมูลของเครื่องคอมพิวเตอร์ เชื่อมเข้าหากันด้วยครอสคอนเน็คท์ความเร็วสูง (AXI) บัสความเร็วสูง (AHB) และบัสอุปกรณ์ต่อพ่วง (APB)
- การทำงานจะใช้กลไกการติดต่อกับอุปกรณ์อินพุท/เอาท์พุท เรียกว่า Memory Mapped IO, กลไกการทำ Interrupt และการทำ Direct Memory Access ร่วมกับ อุปกรณ์อินพุท เอาท์พุท ต่างๆ เช่น วงจร **GPIO** อุปกรณ์เชื่อมต่อกับเครือข่ายและอุปกรณ์เก็บรักษาข้อมูล เป็นต้น
เชื่อม LED/ switch บลาก