

องค์ประกอบคอมพิวเตอร์และภาษาแอสเซมบลี: กรณีศึกษา Raspberry Pi

บทที่ 3 ฮาร์ดแวร์และซอฟต์แวร์ของคอมพิวเตอร์

ผศ.ดร.สุรินทร์ กิตติธรรมกุล

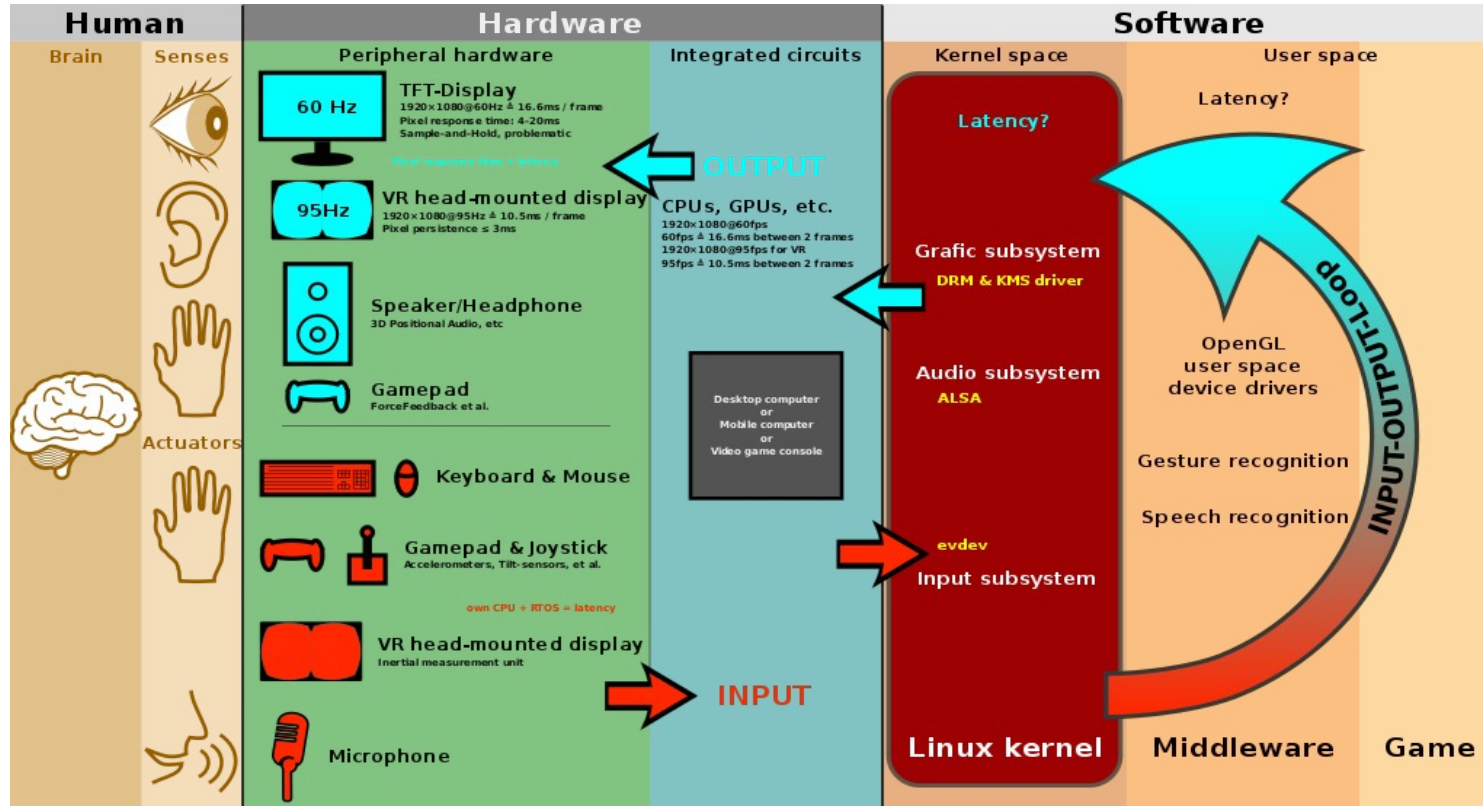
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สารบัญ

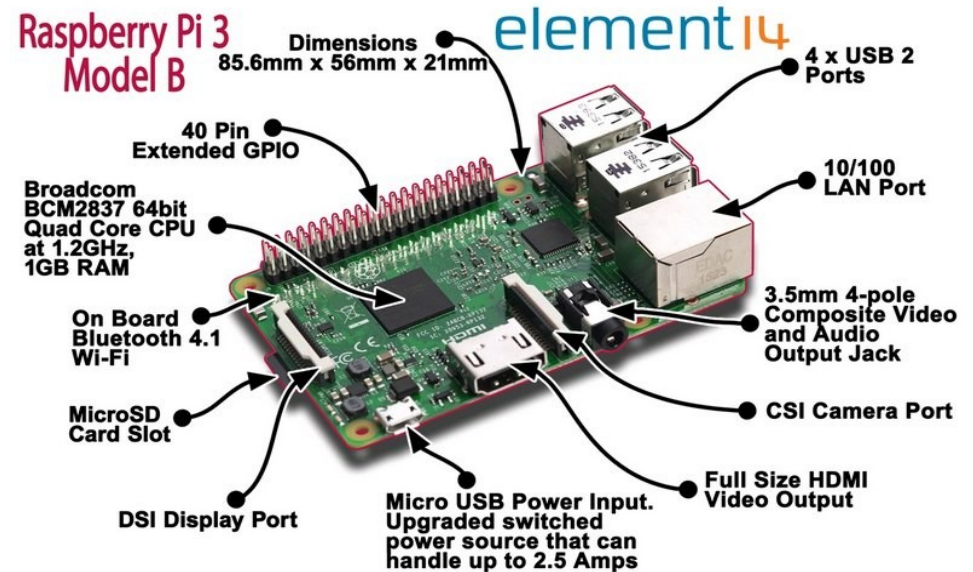
- 3.1. ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์
- 3.2 การพัฒนาซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์
- 3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์
- 3.4 สรุปท้ายบท
- 3.5 คำถามท้ายบท

Human+เครื่องคอมพิวเตอร์ = Human+HW+SW

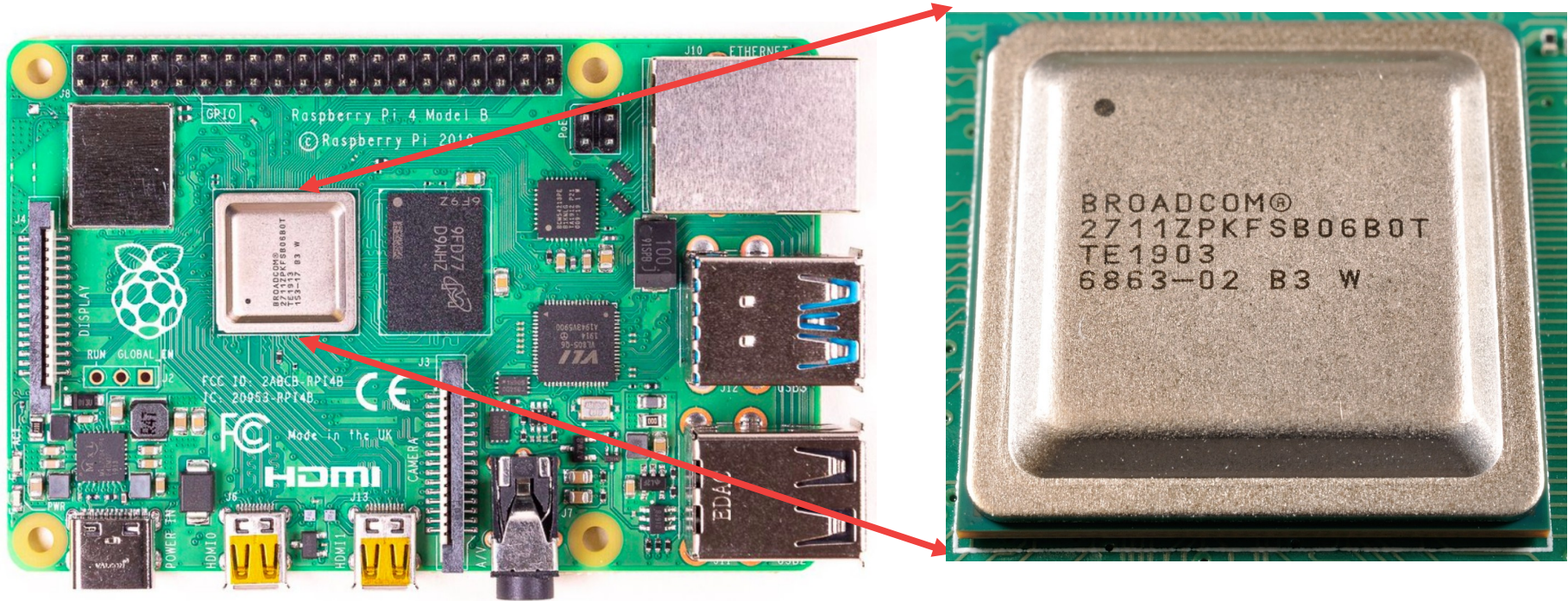


3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์: บอร์ด Pi3

- Processor (multi-core => many core)
- Main Memory (DDR2 SDRAM 1GigaByte)
- Storage (Micro SD Card)
- Input/Output
 - HDMI Monitor, LCD & Camera modules
 - USB, , Ethernet LAN 10/100 Mbps
 - WiFi 802.11n, Bluetooth v.4.1
 - Input/Output Devices
 - USB Keyboard, Mouse, Touch Pad, ...



3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์: บอร์ด Pi4



3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์

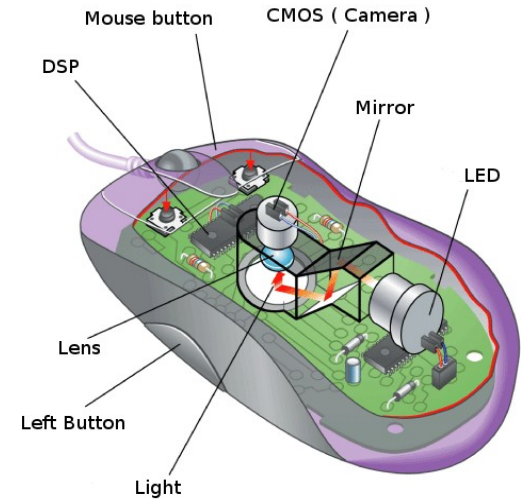
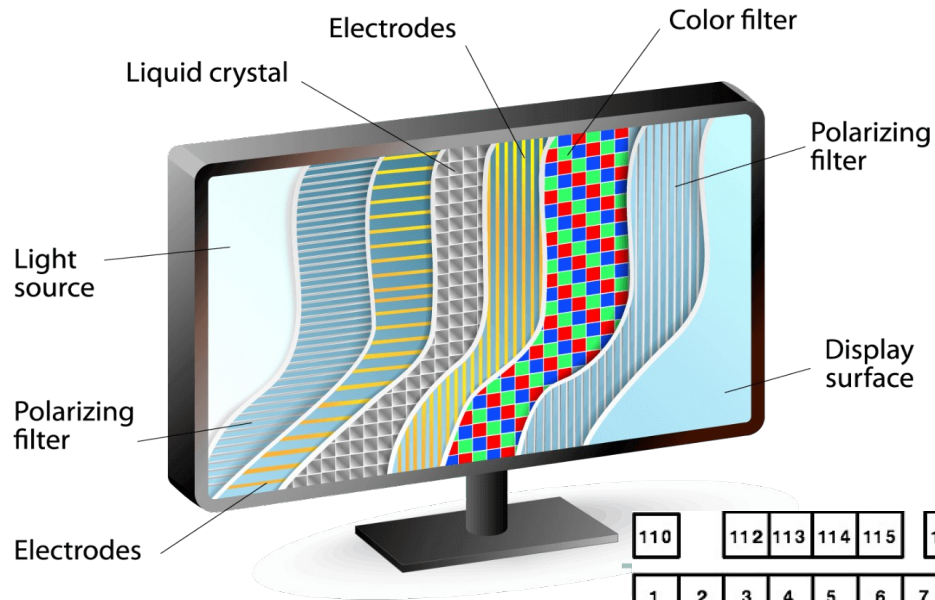
ตารางที่ 3.1: ตารางสรุปข้อมูลด้านฮาร์ดแวร์และซอฟต์แวร์ของบอร์ด Pi3 โมเดล B และลิงค์เชื่อมโยงไปยังหัวข้อที่แสดงรายละเอียดในบทต่างๆ

โมดูล	ภายในชิพ (On chip) BCM2837	ในหัวข้อที่
BCM2837	SoC ผลิตโดยบริษัท Broadcom ประกอบด้วย	3.1.1
CPU	Quad (4)-Core ARM Cortex-A53 ความถี่ 1.2 GHz	
GPU	Dual (2) VideoCoreIV ความถี่ 400 MHz	
จอ LCD	สาย HDMI เวอร์ชัน 1.3 & 1.4 (ภาพและเสียง)	6.1
จอ LCD	สาย Display Serial Interface (DSI) 15 ขา	6.2
	ประกอบด้วยสัญญาณข้อมูล 2 คู่ สัญญาณคล็อก 1 คู่	
กล้องขนาดเล็ก	สาย Camera Serial Interface (CSI) 15-ขา	6.3
	ประกอบด้วยสัญญาณข้อมูล 2 คู่ สัญญาณคล็อก 1 คู่	
จอทีวี	สัญญาณคอมโพสิทวิดีโอ PAL/NTSC	6.5
และเสียง	แจ็คขนาด 3.5 มม ชนิด 4 ขั้ว	6.4
GPIO	ขั้วต่อชนิด 2.54 มม 40 ขา	6.11
	ประกอบด้วย GPIO 27 ขา +3.3 และ +5V โวลท์	

3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์

อุปกรณ์	ภายนอกชิพ (Off chip) BCM2837	ในหัวข้อที่
ชิพ SDRAM	หน่วยความจำชนิด DDR2 ความจุ 1 จิกะไบต์ ชนิดประหยัดพลังงาน	3.1.2, 5.5
ชิพ USB	ชิพ USB 2.0 จำนวน 4 พอร์ต	6.6
ชิพ Ethernet	เชื่อมต่ออินเทอร์เน็ตผ่านสาย ด้วยอัตรา 10/100 Mbps	6.7
ชิพ WiFi และ Bluetooth	เชื่อมต่ออินเทอร์เน็ตไร้สาย IEEE 802.11 b/g/n อัตราเร็วสูงสุด 150Mbps การเชื่อมต่อไร้สายเวอร์ชัน 4.1	6.8
แหล่งจ่ายไฟ การ์ด SD	ซ็อกเก็ตชนิด microUSB ขนาด 5 โวลต์ 2.5 แอมแปร์ อุปกรณ์เก็บรักษาข้อมูลความจุ 4-16 จิกะไบต์	6.14 3.1.4, 7.3
ซอฟต์แวร์	ระบบปฏิบัติการบูทจากการ์ด SD รองรับ Raspbian Linux Windows 10 IoT และอื่นๆ	3.2.1

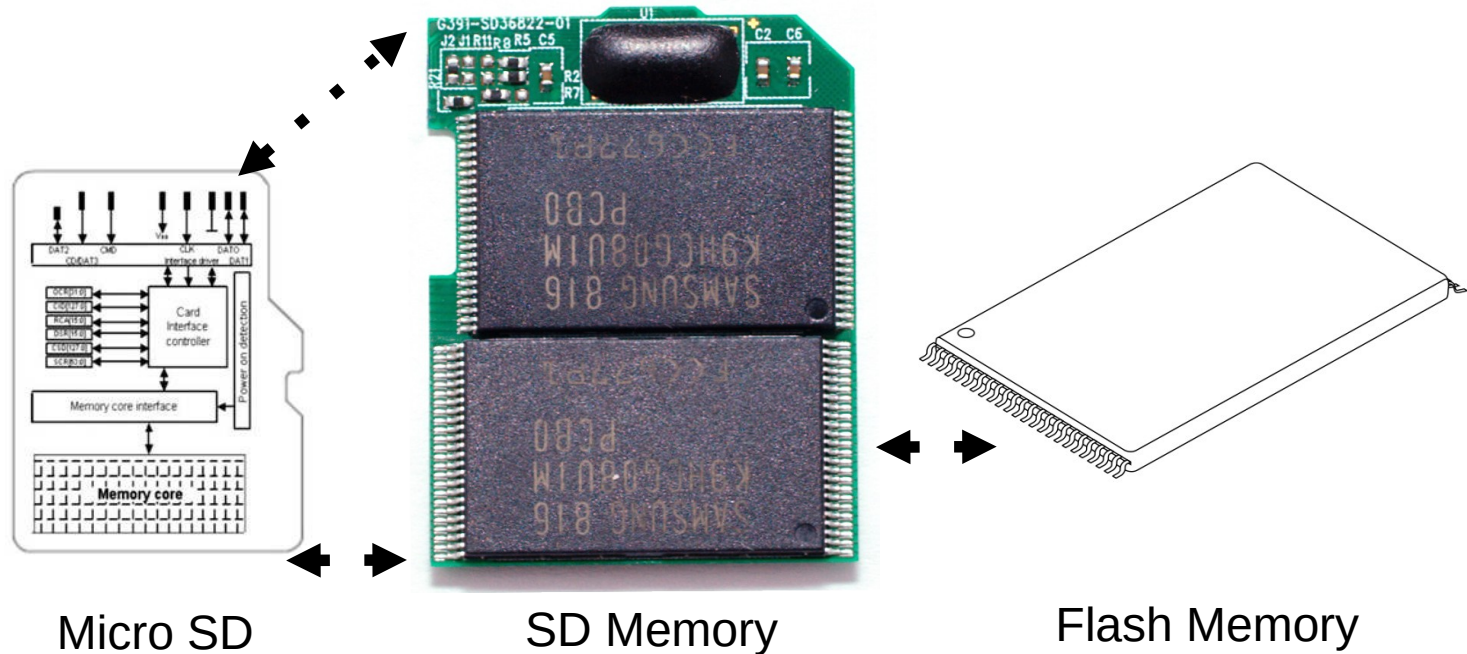
3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์: Input/Output



110	112 113 114 115				116 117 118 119				120 121 122 123				124 125 126								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	75	80	85	90	95	100	105
16	17	18	19	20	21	22	23	24	25	26	27	28	(29)		76	81	86	91	96	101	106
30	31	32	33	34	35	36	37	38	39	40	41	42	43					92	97	102	107
44	45	46	47	48	49	50	51	52	53	54	55	56	57				83	93	98	103	108
58		60	131	61	132	133	62			64	79	84	89					(94)	99	104	(109)

106-Key Keyboard Position Codes

3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์: Storage

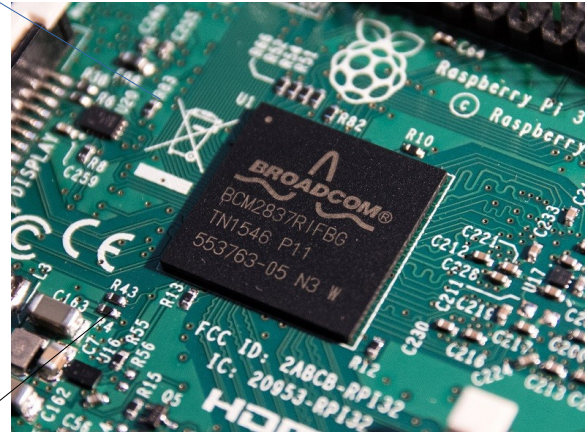
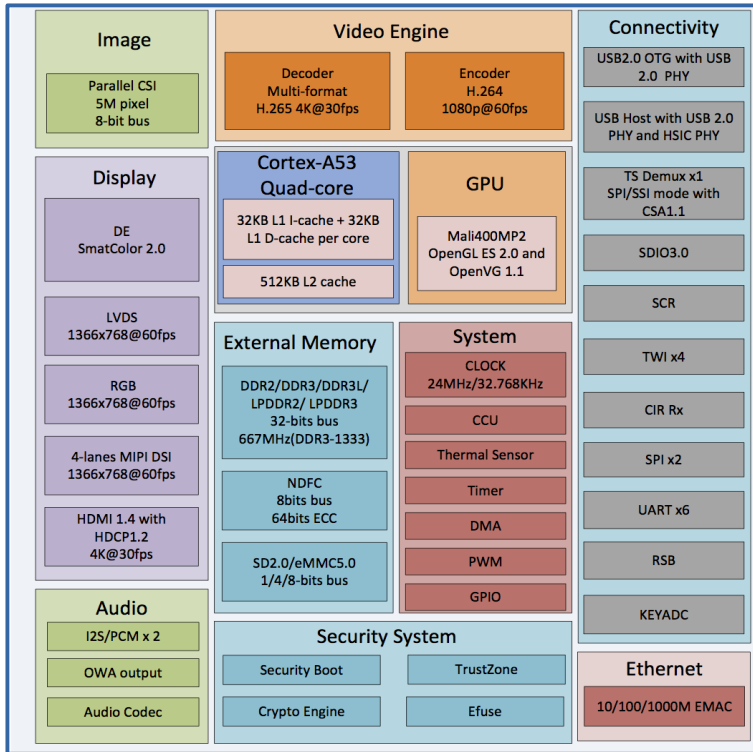


3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์: SDRAM LPDDR2



ผลิตโดยบริษัท Elpida ประเทศญี่ปุ่น
Low Power ไฟเลี้ยงต่ำประมาณ 1.2 โวลต์
ความถี่สัญญาณคล็อก – 400 MHz
READ/WRITE Latency สามารถตั้งจำนวนเท่า
ของคาบเวลาคล็อก = Access Time
สามารถอ่าน/เขียนข้อมูล 4, 8 และ 16
ตำแหน่งต่อเนื่องได้
รายละเอียดเพิ่มเติมในบทที่ 5

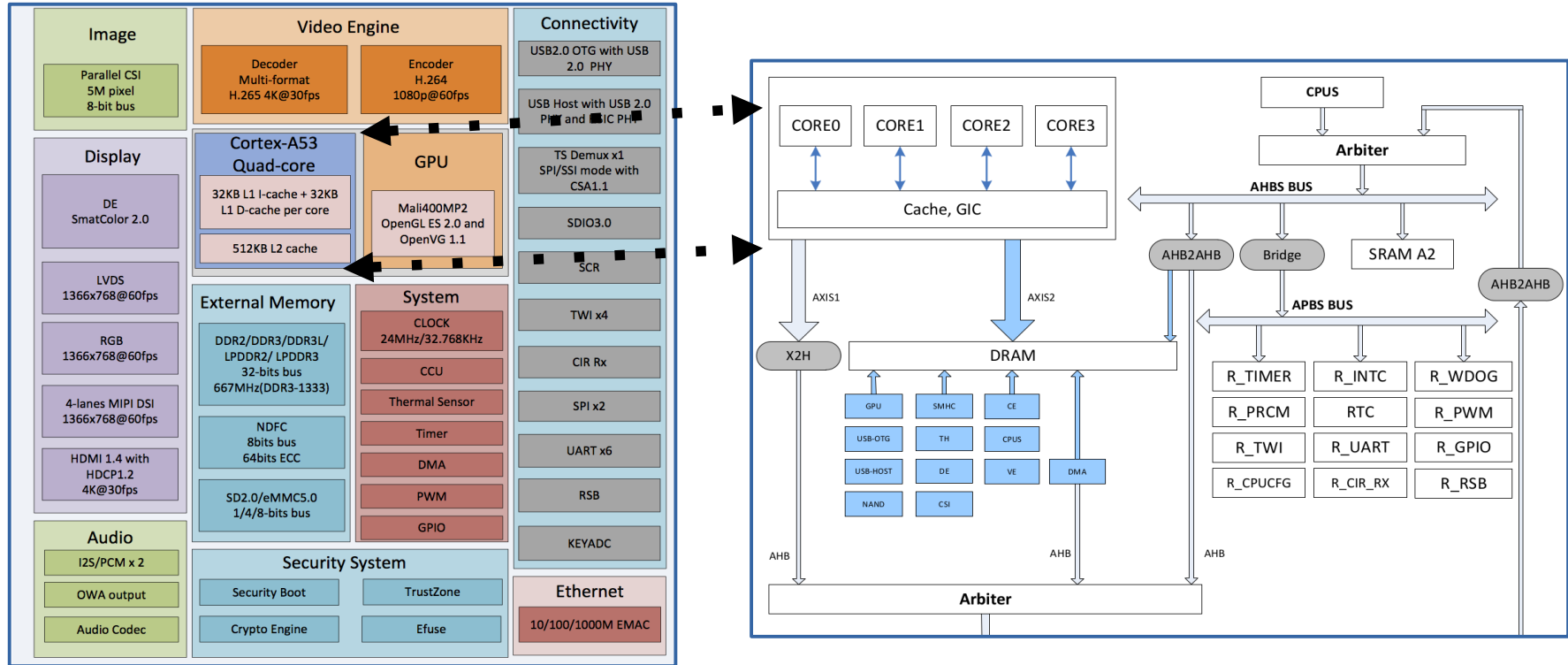
3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์: SoC BCM2837



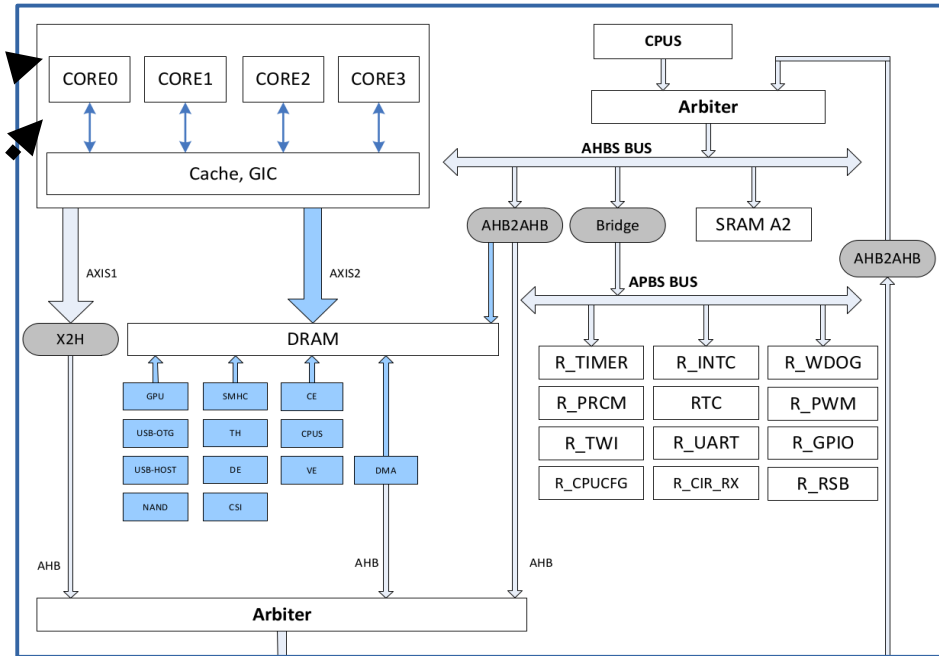
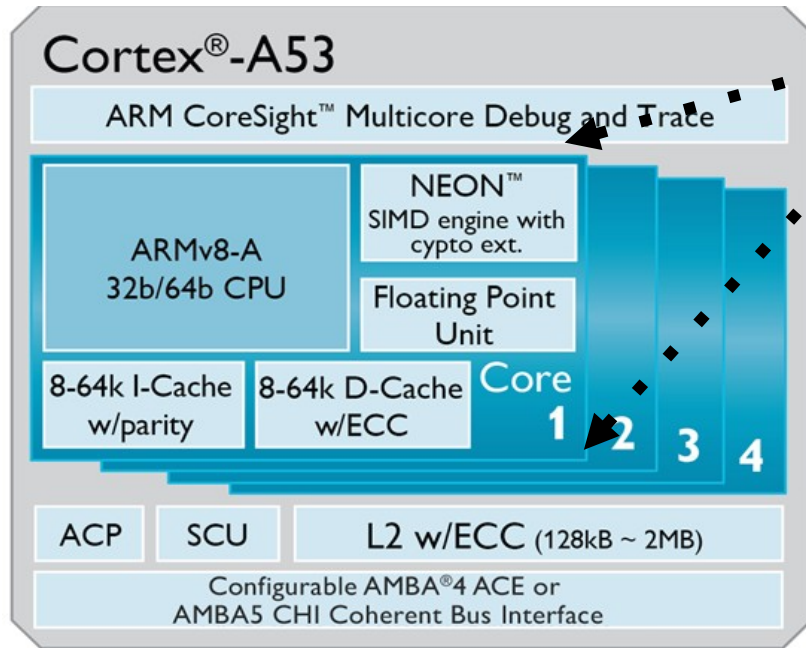
ชิพ BCM2837

ชิพที่คล้ายกับ BCM2837

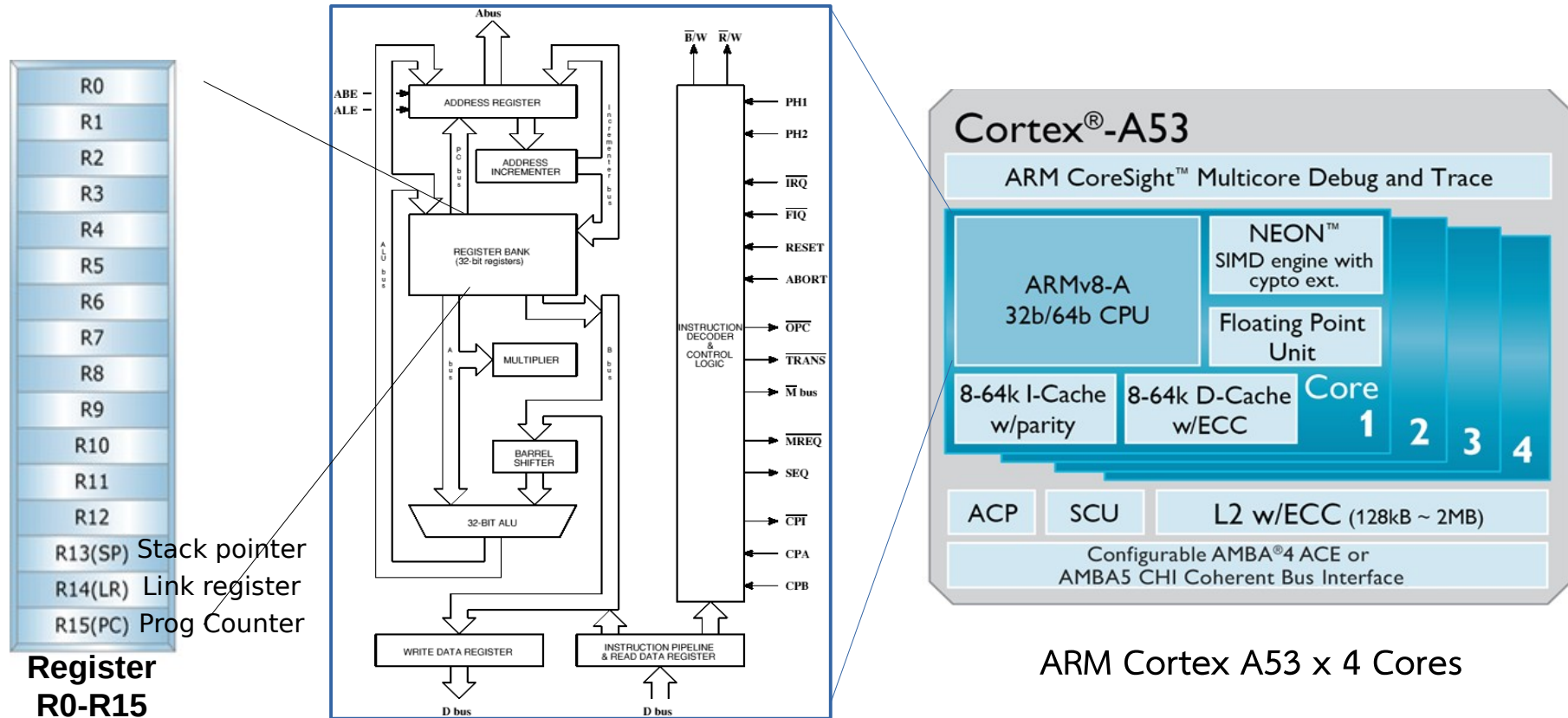
3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์: ซีพียู ARM Cortex A53 x4



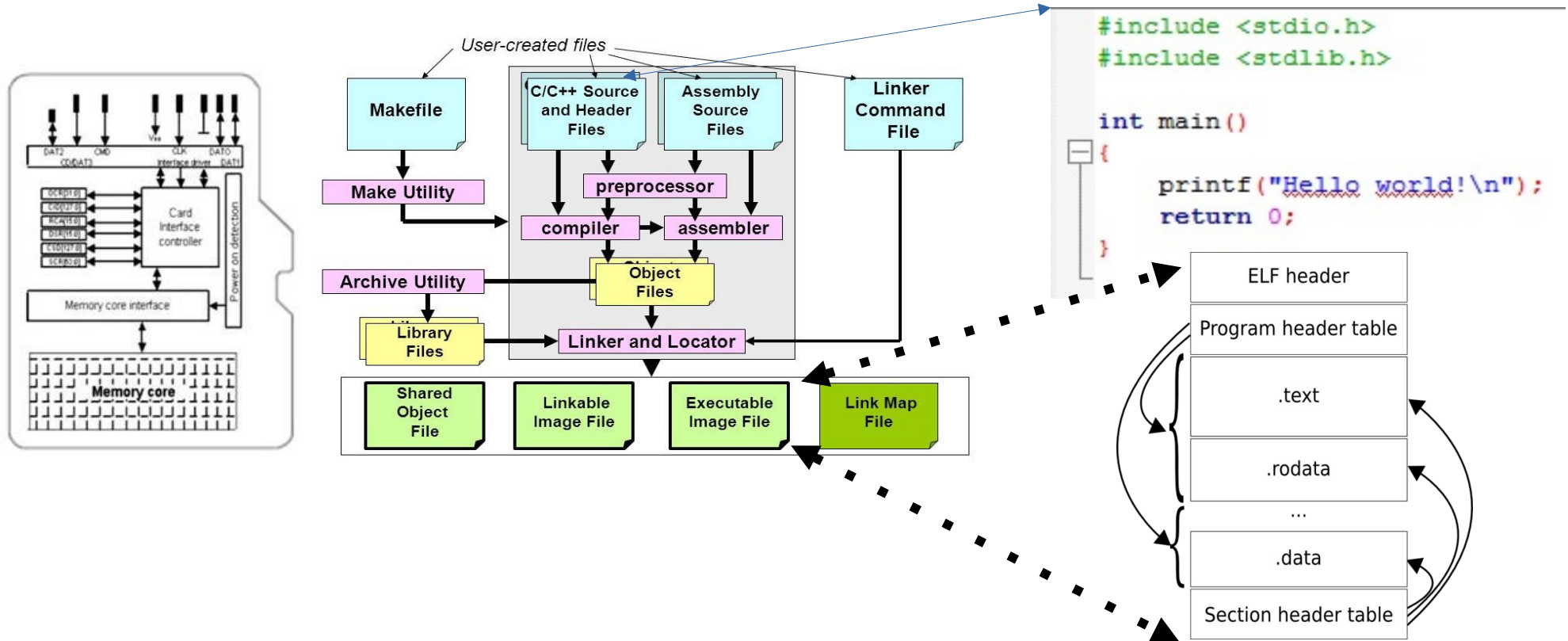
3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์: ซีพียู ARM Cortex A53 x4



3.1 ฮาร์ดแวร์ของเครื่องคอมพิวเตอร์: ซีพียู ARM Cortex A53 x4

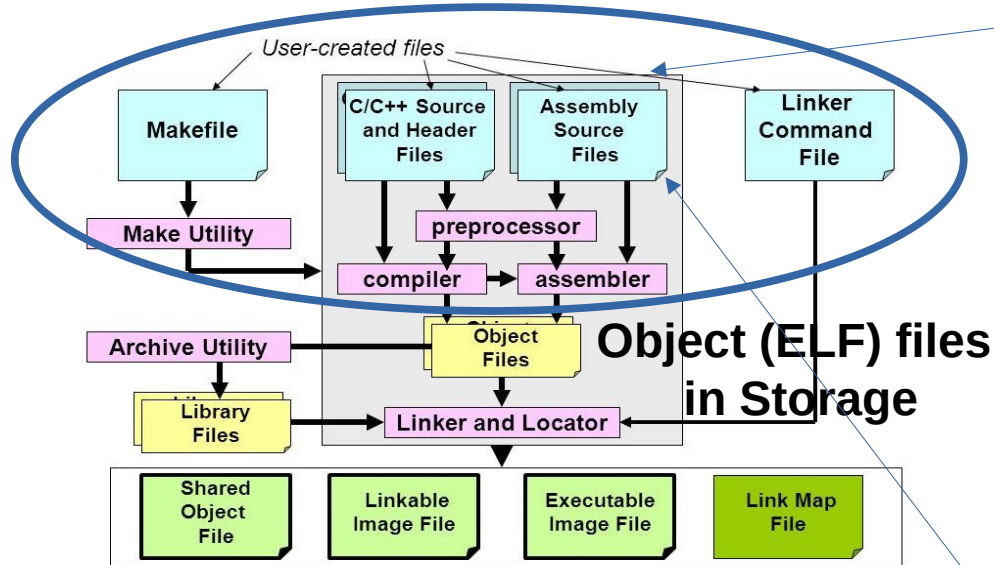


3.2 การพัฒนาซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์: Source Code ภาษา C และภาษาแอสเซมบลี



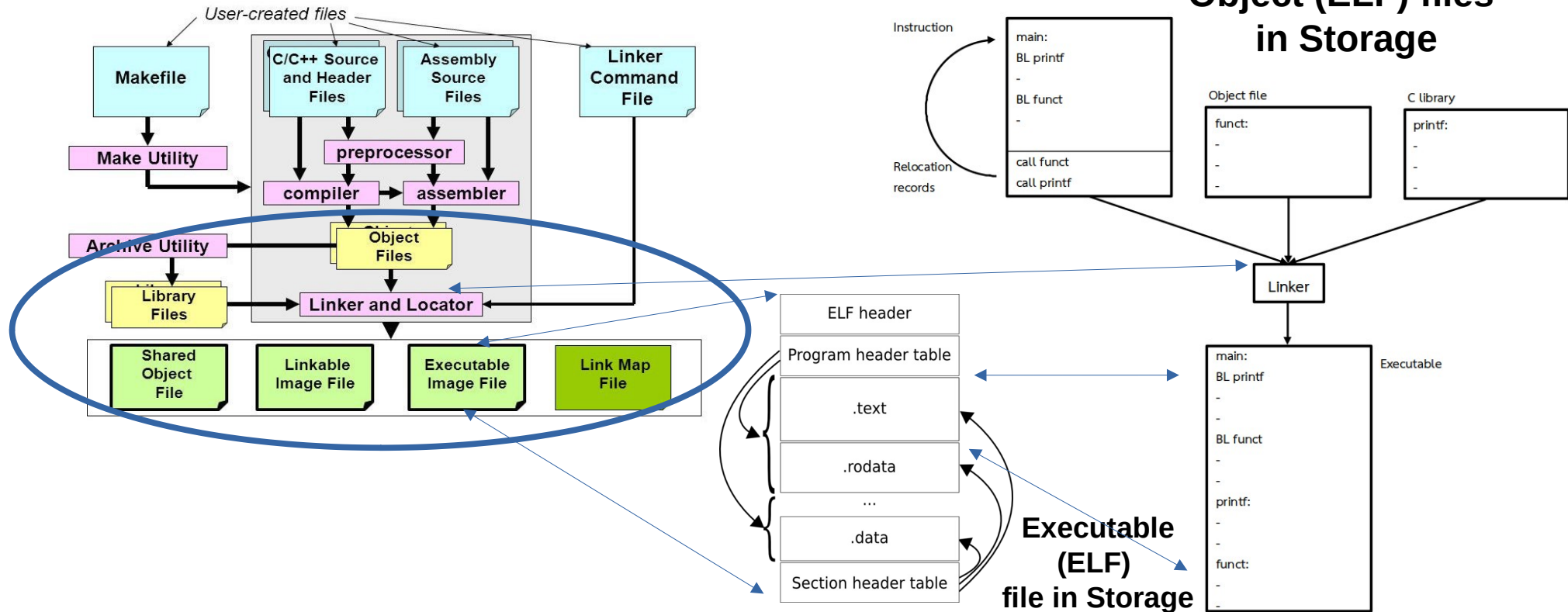
3.2 การพัฒนาซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์:

Source Code ภาษา C และภาษาแอสเซมบลี

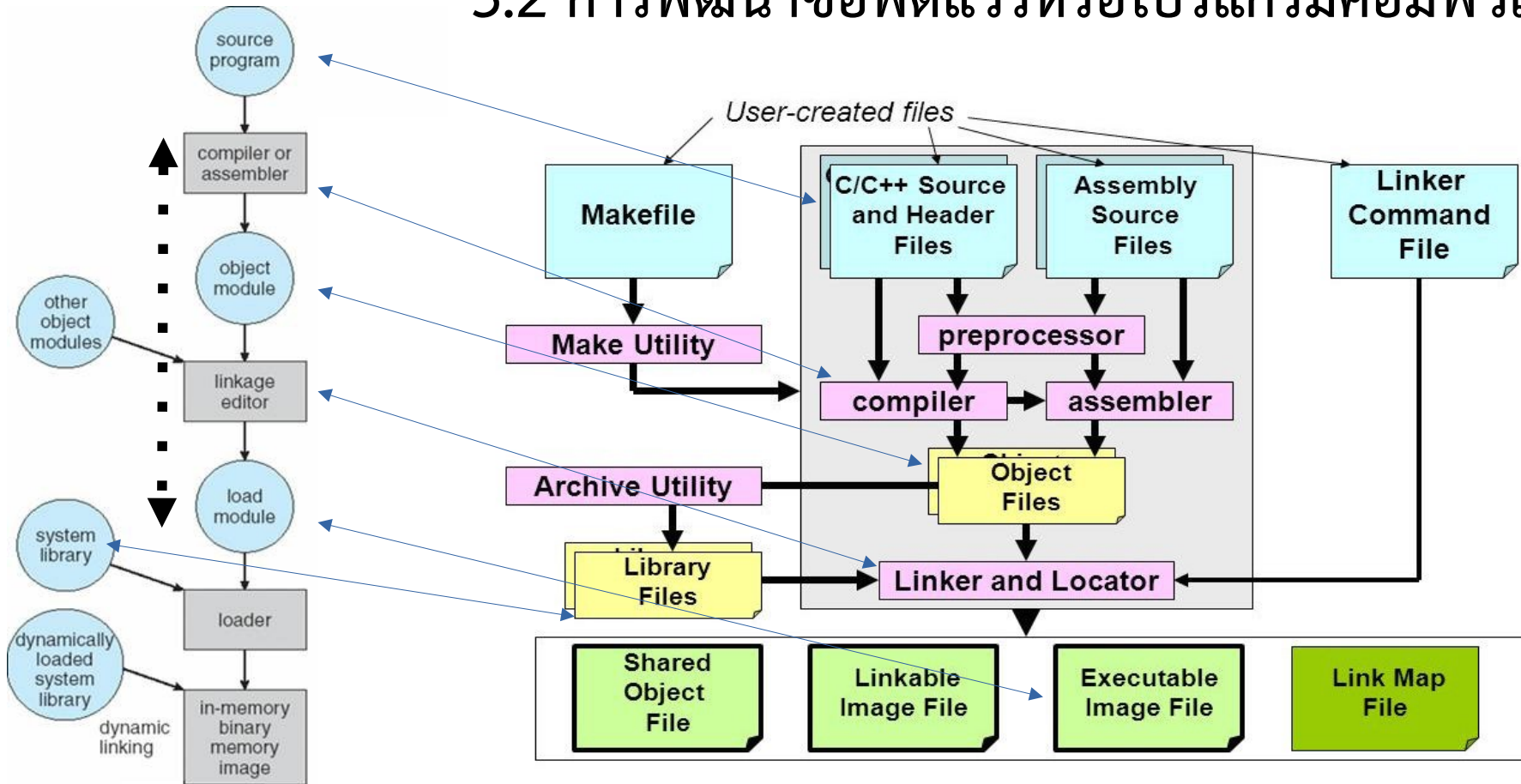


บรรทัดที่	เลเบล	คำสั่ง	รีจิสเตอร์ หรือ แอดเดรส หรือ เลเบล หรือ ค่าคงที่
1		.text	
2		.global main	
3	main:		
4		LDR	R1, =M
5		LDR	R1, [R1]
6		LDR	R2, =POINTR
7		MOV	R0, #0
8	LOOP:	LDR	R3, [R2], #4
9		ADD	R0, R0, R3
10		SUBS	R1, R1, #1
11		CMP	R1, #0
12		BGT	LOOP
13		LDR	R4, =SUM
14		STR	R0, [R4]
15		BX	LR
16		.data	
17	SUM :	.word	#0
18	M:	.word	#4
19	NUM:	.word	3, 5, 7, 9
20	POINTR:	.word	NUM

3.2 การพัฒนาซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์: Linking

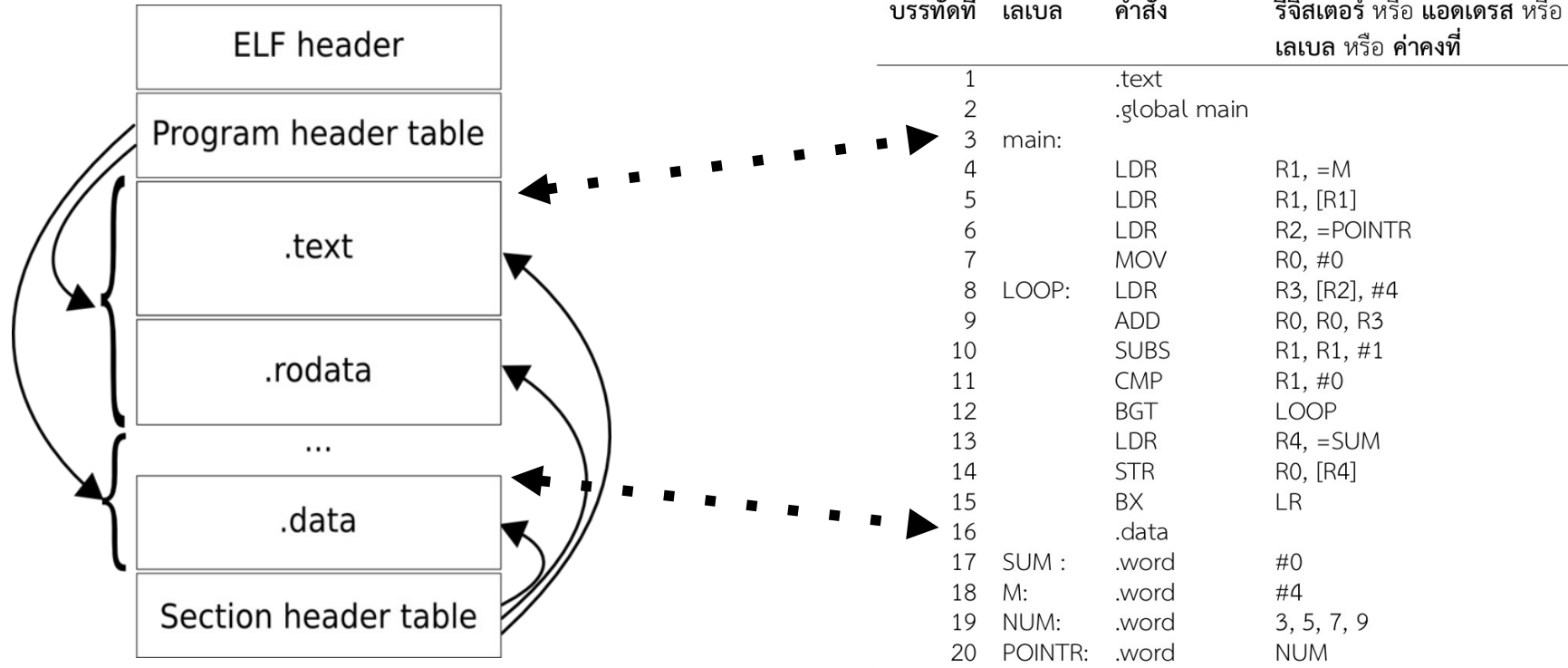


3.2 การพัฒนาซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์

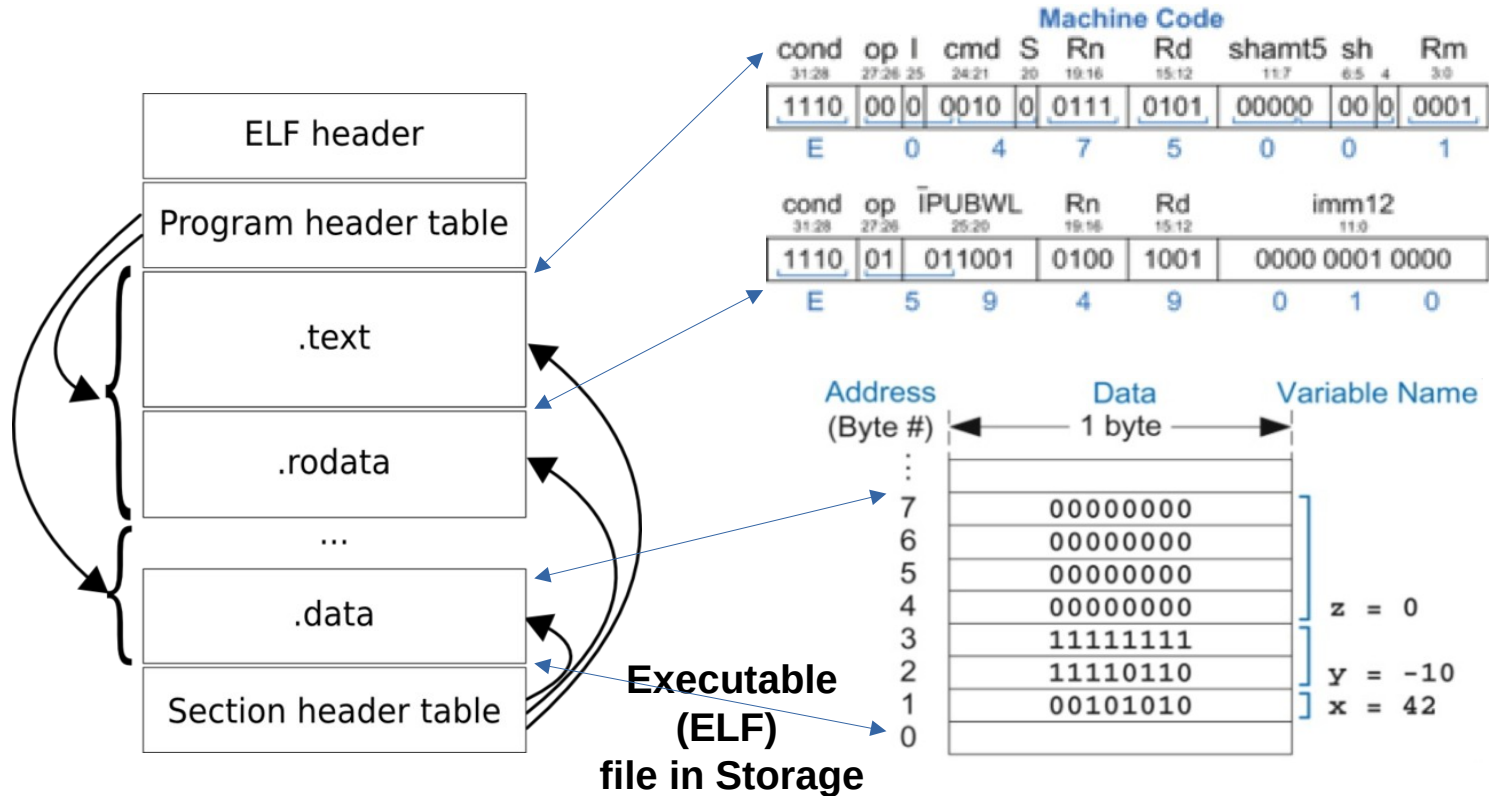


3.2 การพัฒนาซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์:

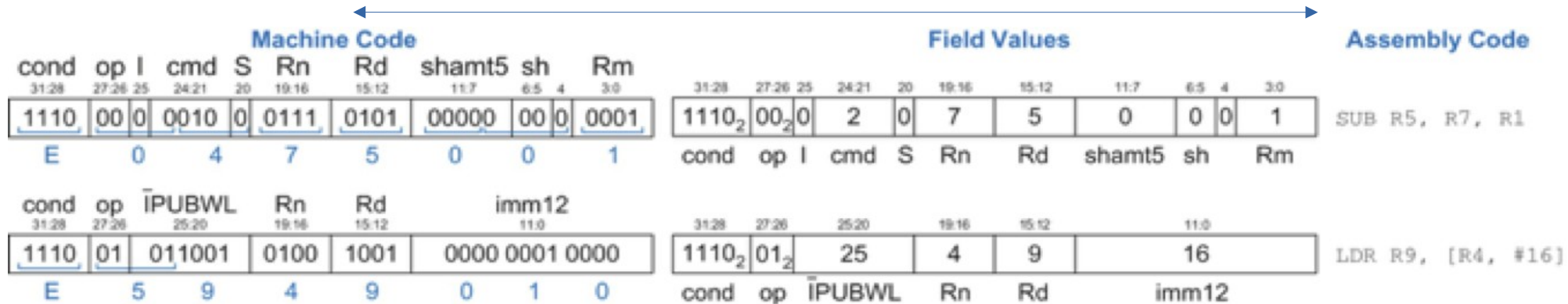
Source Code ภาษา C และภาษาแอสเซมบลี



3.2 การพัฒนาซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์: ภาษาเครื่อง vs ภาษาแอสเซมบลี

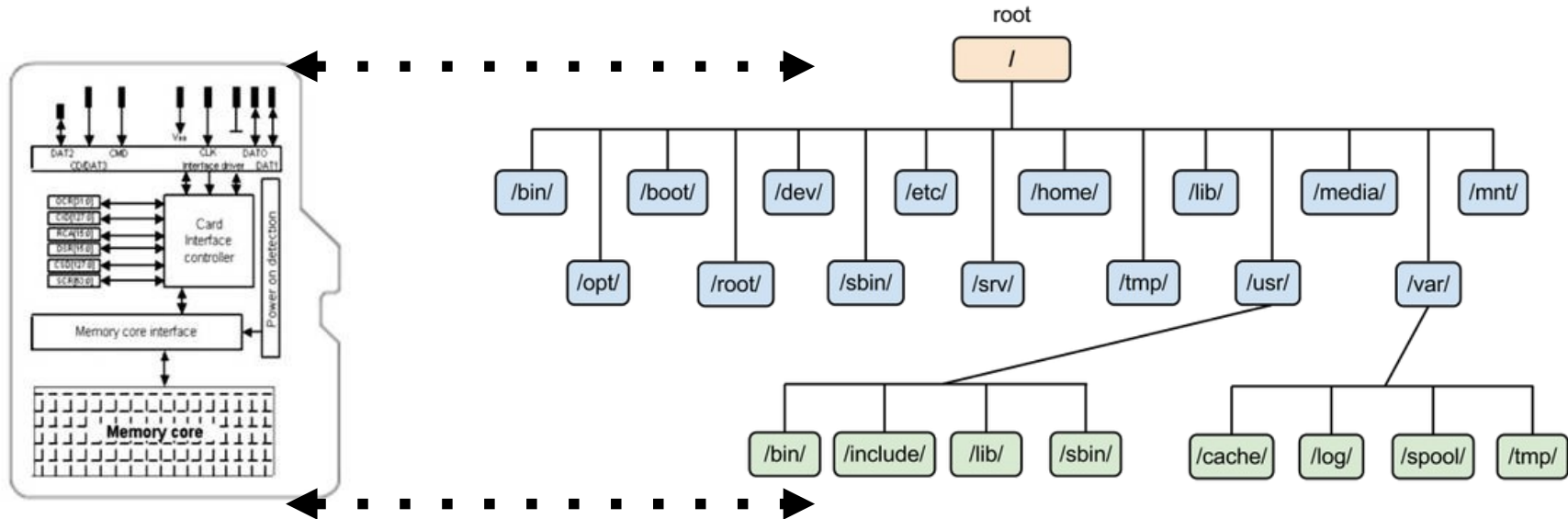


3.2 การพัฒนาซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์: ภาษาเครื่อง vs ภาษาแอสเซมบลี



3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์

เมื่อบูทระบบสำเร็จ เคอร์เนลหรือโปรแกรมหลักของระบบปฏิบัติการจะครอบครองฮาร์ดแวร์ทั้งหมด และทำงานร่วมกับไฟล์ซอฟต์แวร์และไฟล์ข้อมูลต่างๆ ภายในการ์ดหน่วยความจำ SD ภายในการ์ดหน่วยความจำ SD นี้จะแบ่งโครงสร้างการจัดเก็บไฟล์ต่างๆ ตามระบบปฏิบัติการลินุกซ์ เรียกว่า **ไดเรกทอรี** ในรูปที่ 3.9 ผู้อ่านสามารถทำความเข้าใจไดเรกทอรีตามลำดับความสำคัญ ดังนี้

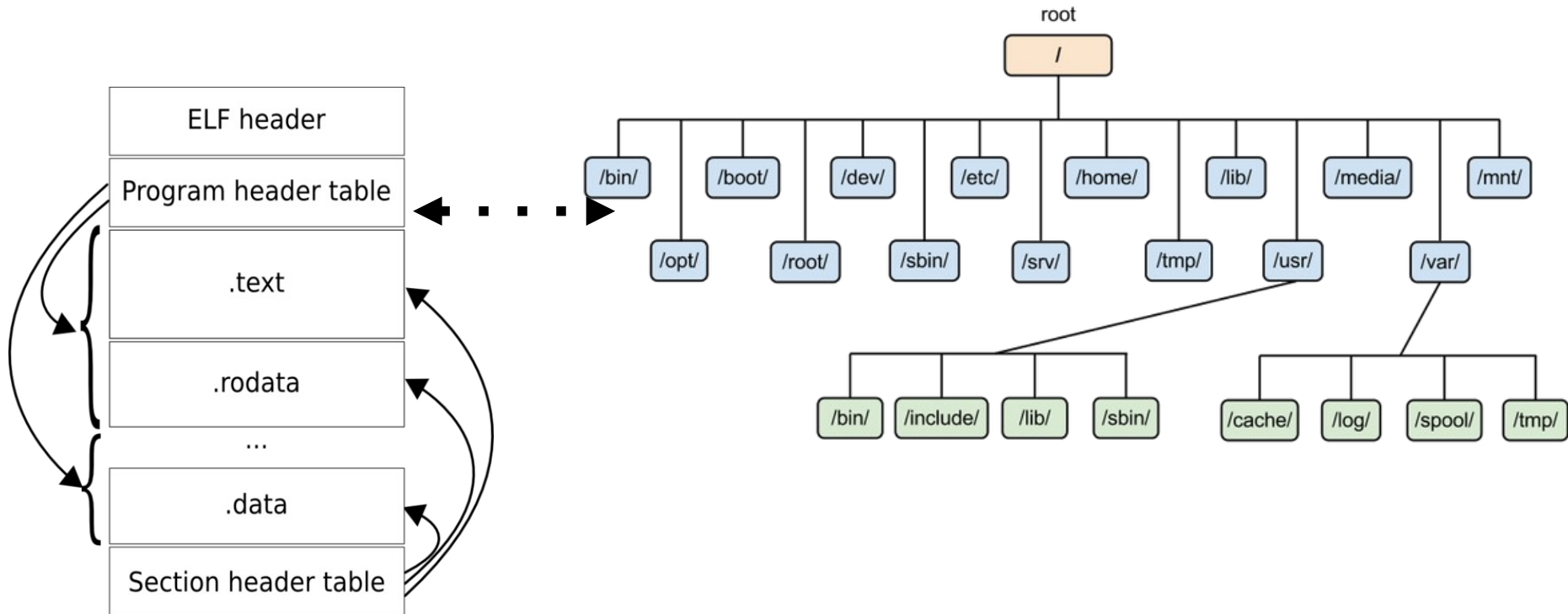


3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์

การทำงานของคอมพิวเตอร์จะเริ่มต้นขึ้นเมื่อผู้ใช้กดปุ่มเปิดเครื่องเพื่อจ่ายไฟเลี้ยงให้กับคอมพิวเตอร์ ระบบฮาร์ดแวร์จะเริ่มทำงานเพื่อ**ตรวจสอบ**และเตรียมความพร้อม หลังจากนั้น ขั้นตอนการทำงานของซอฟต์แวร์ระบบและซอฟต์แวร์ประยุกต์สามารถสรุปได้ดังนี้

1. การบูต (Boot) ระบบปฏิบัติการจากอุปกรณ์เก็บรักษาข้อมูลเข้าสู่หน่วยความจำหลัก ในหัวข้อที่ [3.3.1](#)
2. ระบบปฏิบัติการโหลดไฟล์ซอฟต์แวร์แอปพลิเคชันจากอุปกรณ์เก็บรักษาข้อมูลเข้าสู่หน่วยความจำหลัก ในหัวข้อที่ [3.3.2](#)
3. ซีพียูเพทซ์คำสั่งภาษาเครื่องจากหน่วยความจำหลักไปปฏิบัติตาม ในหัวข้อที่ [3.3.3](#)
4. ซีพียูอ่าน/เขียนข้อมูลระหว่างหน่วยความจำหลักไปประมวลผล ในหัวข้อที่ [3.3.4](#)
5. ซีพียูเชื่อมต่ออุปกรณ์อินพุตต่างๆ เช่น คีย์บอร์ด เมาส์ เครือข่ายอินเทอร์เน็ต เป็นต้น ในหัวข้อที่ [3.3.5](#)
6. ซีพียูอ่าน/เขียนไฟล์ข้อมูลระหว่างหน่วยความจำหลักและอุปกรณ์เก็บรักษาข้อมูล ในหัวข้อที่ [3.3.6](#)
7. ผู้ใช้ชัตดาวน์ (Shut Down) ระบบปฏิบัติการก่อนปิดเครื่อง ในหัวข้อที่ [3.3.7](#)

3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์: ระบบปฏิบัติการโหลดซอฟต์แวร์ประยุกต์จากไฟล์รูปแบบ ELF



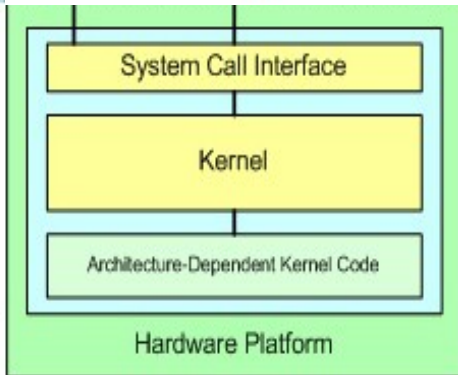
3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์:

โครงสร้างของระบบปฏิบัติการ Linux ในหน่วยความจำหลักหรือ DRAM

- **เคอร์เนลสเปซ (Kernel Space)** เป็นพื้นที่ในหน่วยความจำเสมือนที่จัดสรรโดยเฉพาะสำหรับโปรแกรมเคอร์เนลและองค์ประกอบสำคัญอื่นๆ โครงสร้างภายในของเคอร์เนลสเปซ แบบ Top-Down ประกอบด้วย

- **เคอร์เนล (Kernel)** คือ โปรแกรมหลักภายในระบบปฏิบัติการ ซึ่งปัจจุบันได้มีการพัฒนาอย่างต่อเนื่อง ผู้อ่านสามารถตรวจสอบเวอร์ชันล่าสุดได้ที่ www.kernel.org

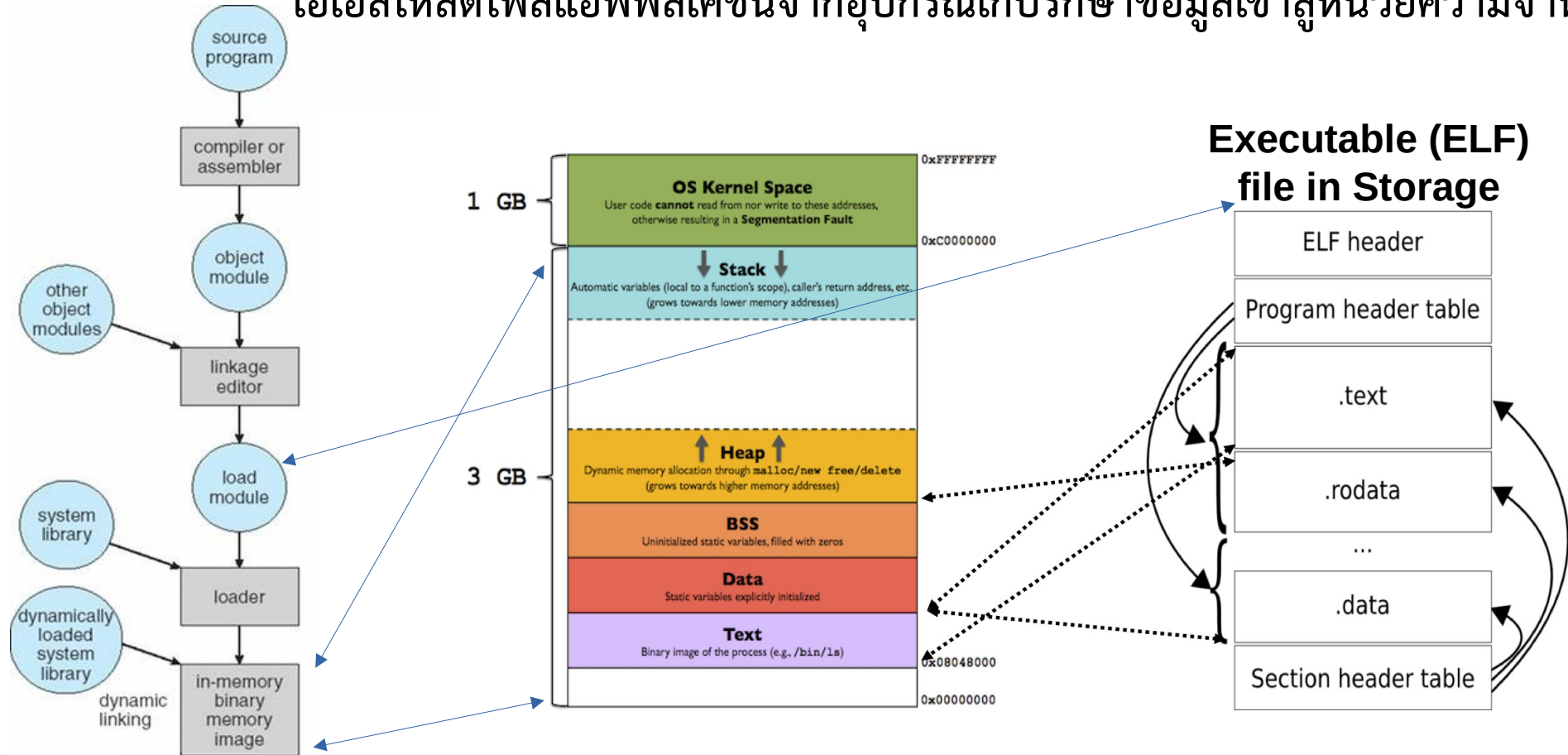
GNU/
Linux



- **ซิสเต็มคอลอินเทอร์เฟส (System Call Interface)** เป็นช่องทางให้ซอฟต์แวร์ประยุกต์เรียกใช้งานเคอร์เนล เพื่อให้เคอร์เนลช่วยส่งงานอุปกรณ์อินพุต/เอาต์พุต การอ่าน/เขียนไฟล์ เป็นต้น
- **คำสั่งภายในเคอร์เนลที่ผูกติดกับสถาปัตยกรรมของฮาร์ดแวร์ (Architecture Dependent Kernel Code)** สำหรับใช้งานเชื่อมต่อกับฮาร์ดแวร์ต่างๆ โดยเฉพาะ ซึ่งคำสั่งส่วนนี้ของเคอร์เนลจะเปลี่ยนแปลงไปตามซีพียูและชิพที่ทำงาน เพื่อให้เคอร์เนลติดต่อกับทรัพยากรเหล่านั้นได้อย่างถูกต้องและมีประสิทธิภาพ คำสั่งและข้อมูลประกอบนี้ เรียกว่า ดีไวส์ไดรเวอร์

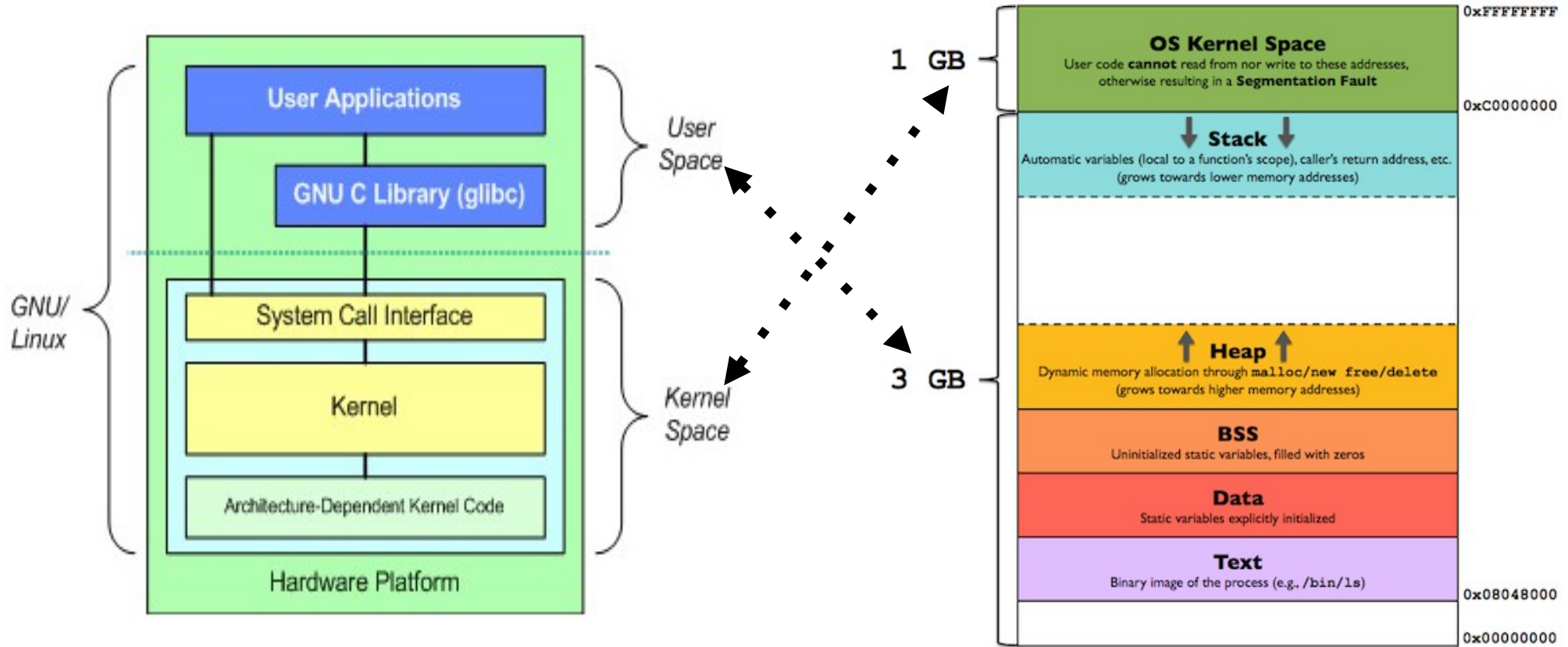
3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์:

โอเอสโหลดไฟล์แอปพลิเคชันจากอุปกรณ์เก็บรักษาข้อมูลเข้าสู่หน่วยความจำหลัก

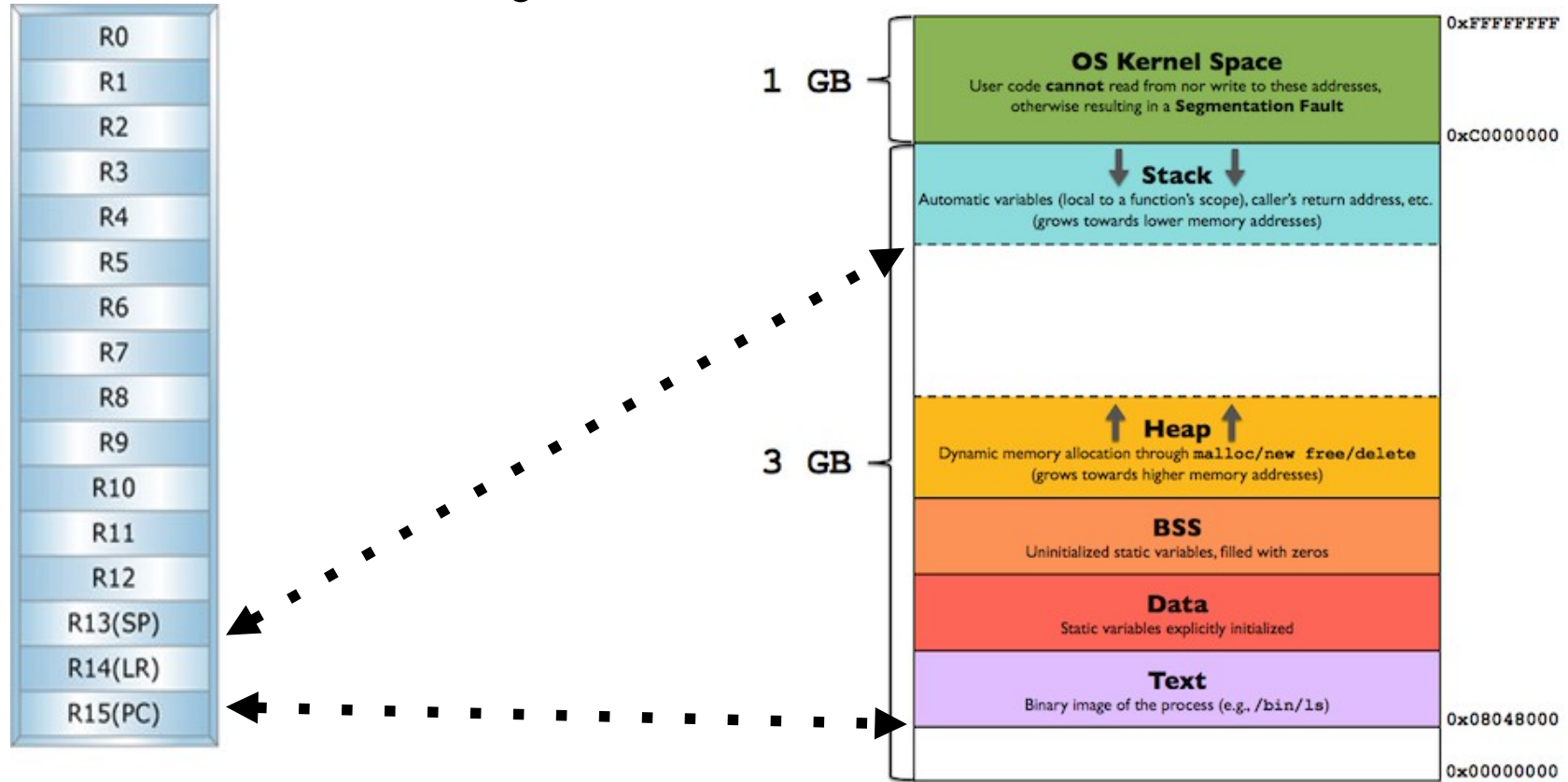


3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์:

โครงสร้างของระบบปฏิบัติการ Linux ในหน่วยความจำหลักหรือ DRAM

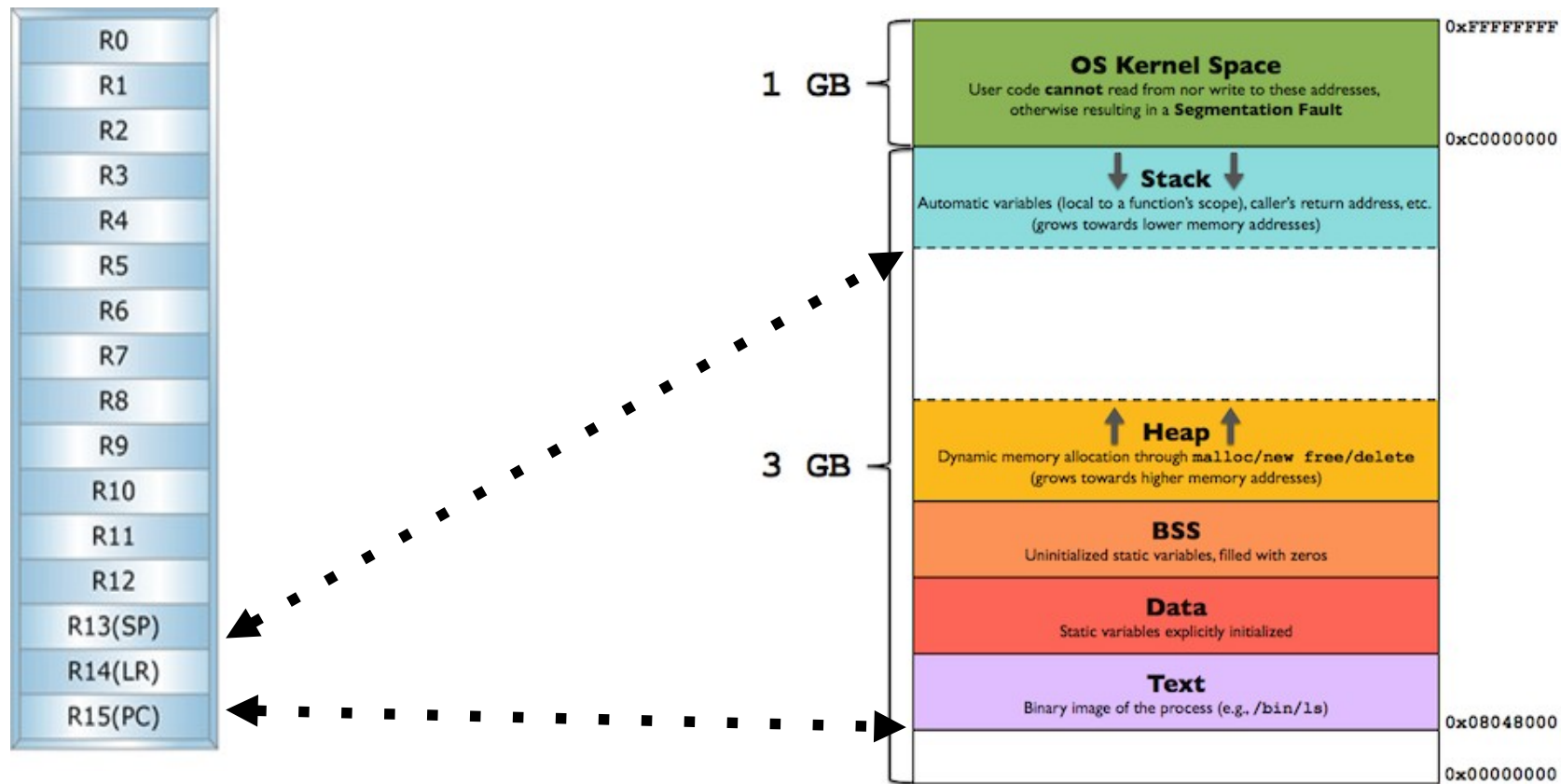


3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์: รีจิสเตอร์สำคัญๆ กับหน่วยความจำหลักหรือ DRAM



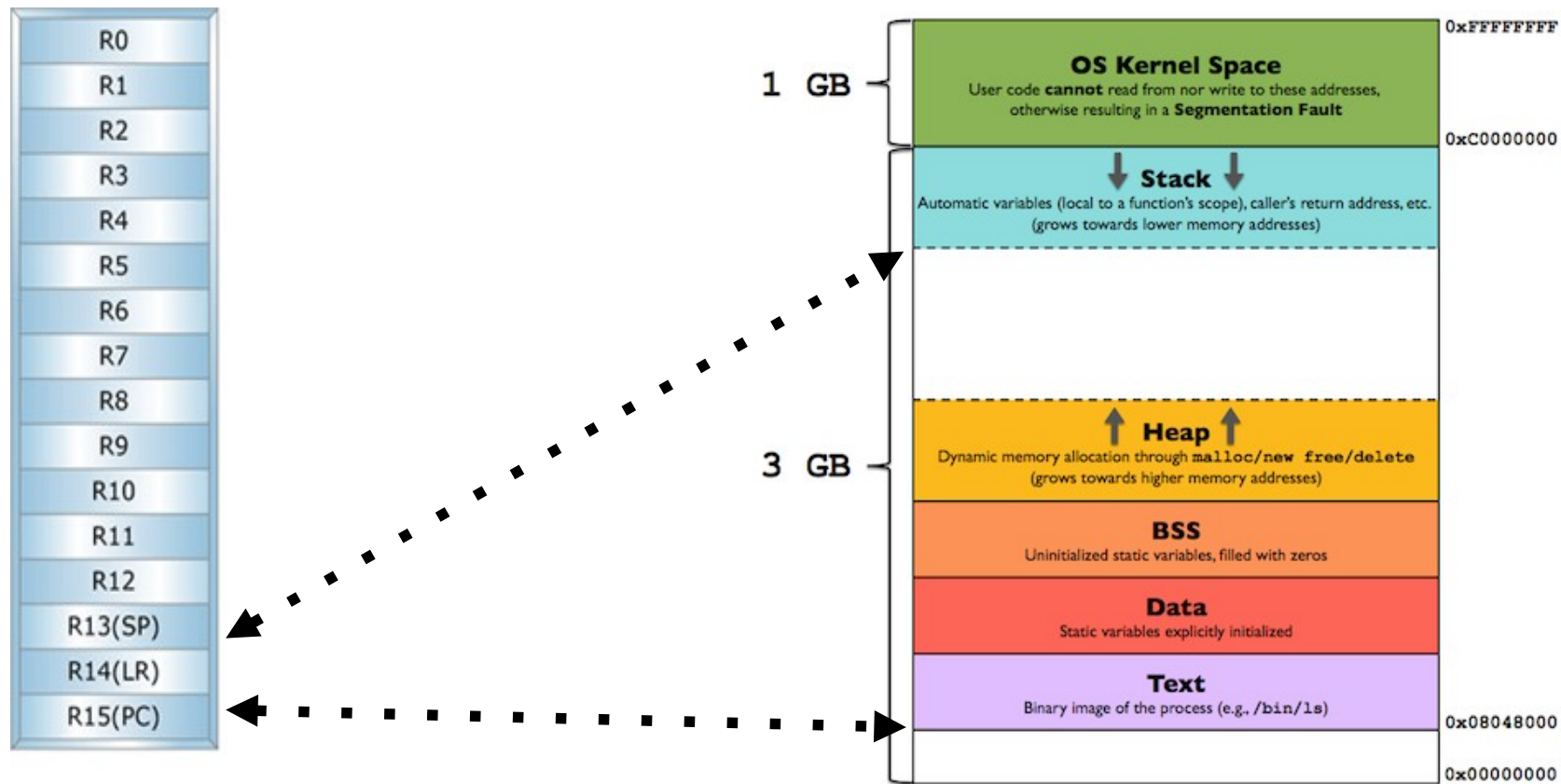
3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์:

ซีพียูเฟตช์ (Fetch) คำสั่งจากหน่วยความจำหลักไปถอดรหัส (Decode) และปฏิบัติตาม (Execute)



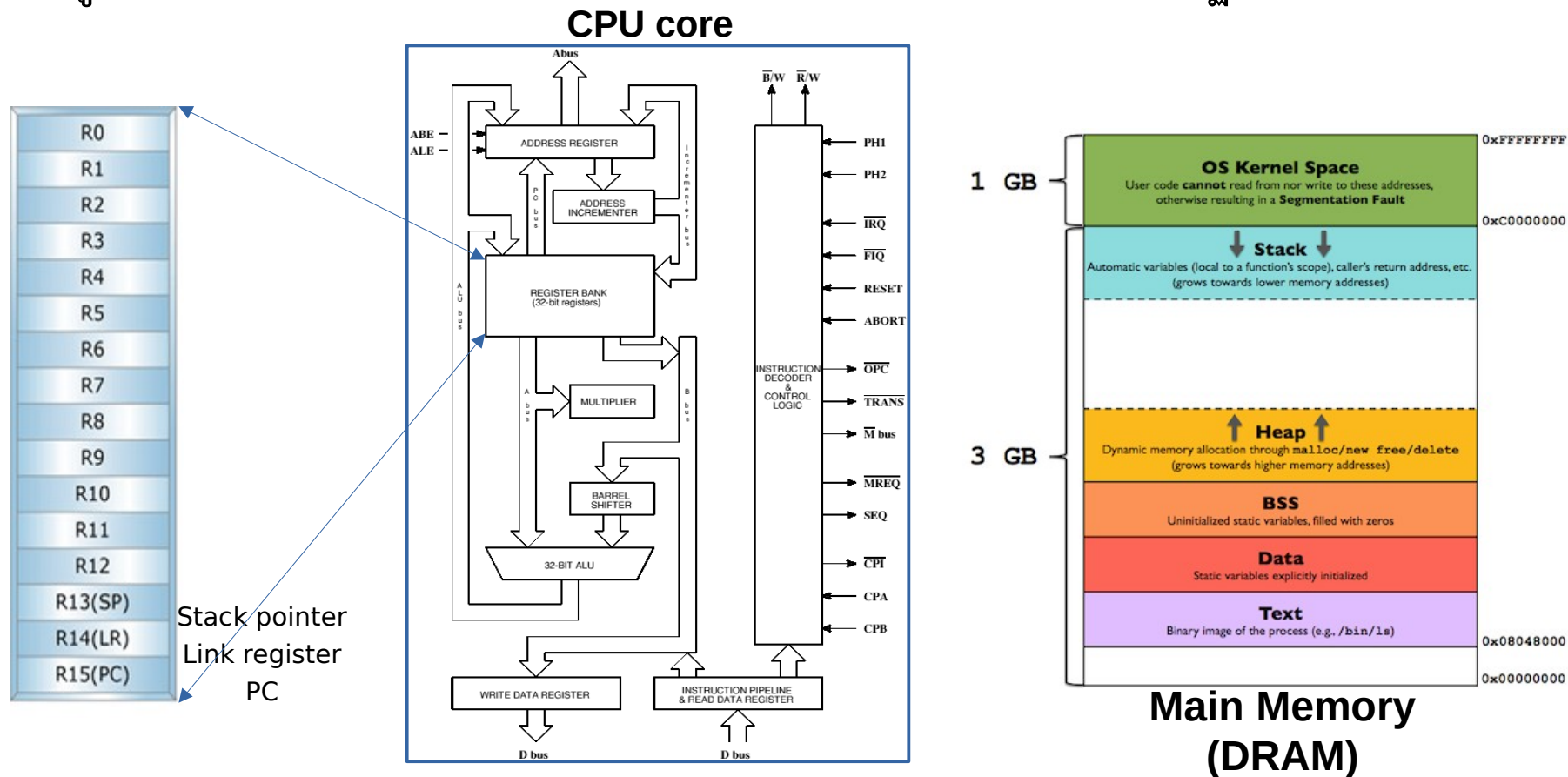
3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์:

ซีพียูเฟตช์ (Fetch) คำสั่งจากหน่วยความจำหลักไปถอดรหัส (Decode) และปฏิบัติตาม (Execute)

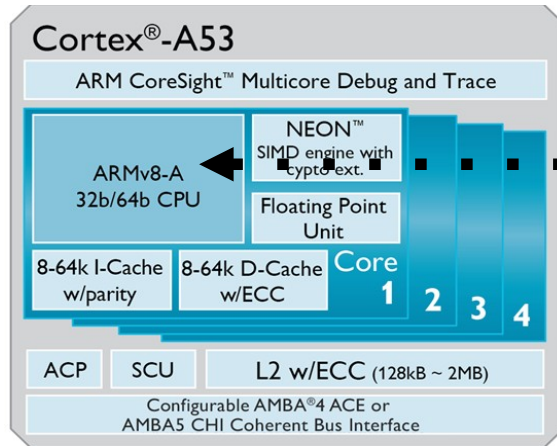


3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์:

ซีพียูเฟทช์ (Fetch) คำสั่งจากหน่วยความจำหลักไปถอดรหัส (Decode) และปฏิบัติตาม (Execute)



3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์: ชนิดของคำสั่งในบทที่ 4

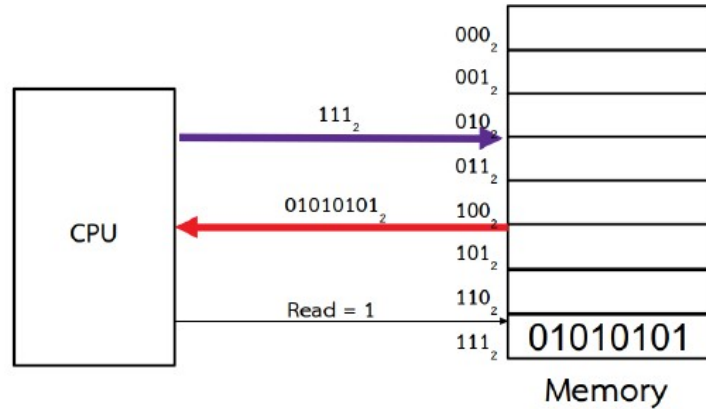


- ชนิดของคำสั่ง แบ่งเป็น

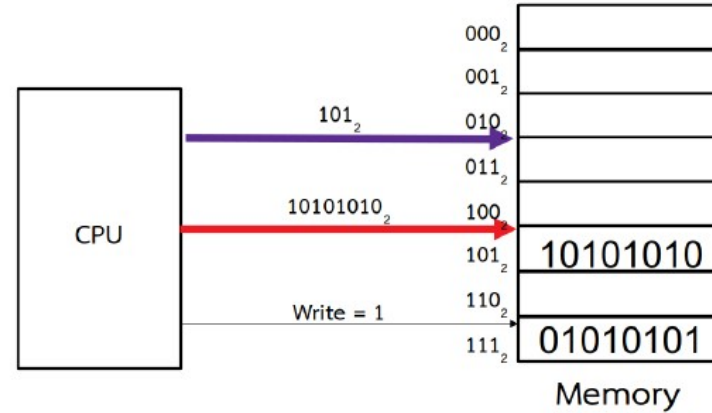
- คำสั่งประกาศและตั้งค่าเริ่มต้นตัวแปร
- คำสั่งการถ่ายโอนข้อมูลระหว่างตัวแปรในหน่วยความจำกับรีจิสเตอร์
- คำสั่งประมวลผลคณิตศาสตร์และตรรกศาสตร์จากข้อมูลในรีจิสเตอร์
- คำสั่งการควบคุมการทำงาน เพื่อการตัดสินใจและการวนรอบทำซ้ำ
- คำสั่งเรียกใช้ฟังก์ชันและรีเทิร์น (Return) กลับ

3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์:

คำสั่ง Load และคำสั่ง Store ค่าของตัวแปรในหน่วยความจำหลักหรือ DRAM



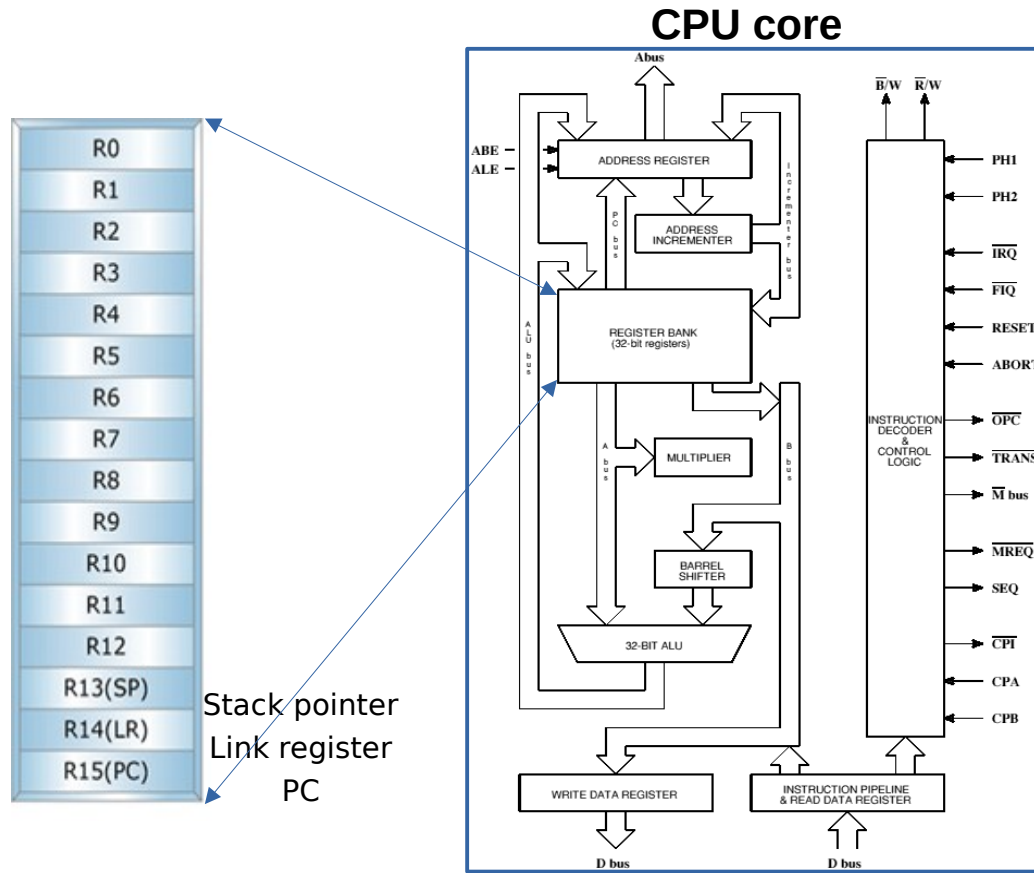
(a)



(b)

รูปที่ 3.17: (a) ขบวนการอ่าน (Load) ข้อมูลจากหน่วยความจำหลักที่ตำแหน่ง 111_2 (b) ขบวนการเขียน (Store) ข้อมูลในหน่วยความจำหลักที่ตำแหน่ง 101_2

3.3 การทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์: คำสั่งคณิตศาสตร์คำนวณค่าของตัวแปรในรีจิสเตอร์



References

- https://www.researchgate.net/figure/Block-Diagram-of-Micro-SD-card_fig6_306236972
- <https://gabrieletolomei.wordpress.com/miscellanea/operating-systems/in-memory-layout/>
- <https://freedompenguin.com/articles/how-to/learning-the-linux-file-system>
- <https://www.techpowerup.com/174709/arm-launches-cortex-a50-series-the-worlds-most-energy-efficient-64-bit-processors>
- https://www.researchgate.net/figure/NVIDIA-Tegra-2-mobile-processor-11_fig1_221634532
- Harris, D. and S. Harris (2013). Digital Design and Computer Architecture (1st ed.). USA: Morgan Kauffman Publishing.
- <https://learn.adafruit.com/resizing-raspberry-pi-boot-partition/edit-partitions>

References

- https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction
- <https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/programmer-s-guide-for-armv8-a>
- https://xdevs.com/article/rpi3_oc/
- https://www.gsmarena.com/a_look_inside_the_new_proprietary_apple_a6_chipset-news-4859.php
- https://www.slideshare.net/kleinerperkins/2012-kpcb-internet-trends-yearend-update/25-Global_Smartphone_Tablet_Shipments_Exceeded
- <https://www.aliexpress.com/item/32329091078.html>
- <https://www.raspberrypi.org/forums/viewtopic.php?t=63750>
- <https://www.youtube.com/watch?v=2ciyXehUK-U>