

แนะนำ ATmega1280

จารุต บุศราทิจ

คณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยราชภัฏเพชรบุรี

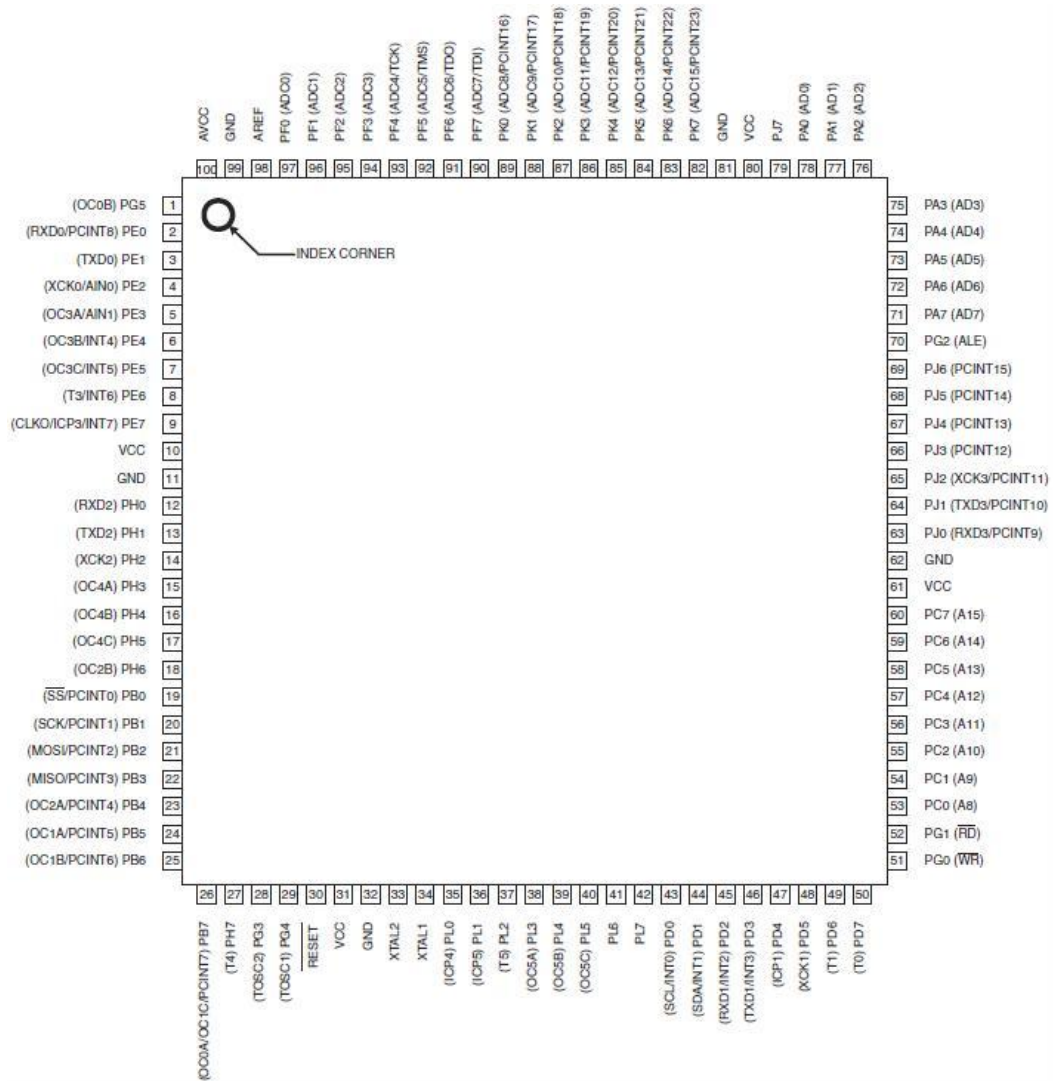
ในบอร์ดรุ่น ET-EASY MEGA1280 (DUINO MEGA) ของบริษัทที่ทีได้เลือกใช้ ชิป ไมโครคอนโทรลเลอร์ ATmega1280 ที่เป็นชิพตระกูล AVR ของบริษัท Atmel รองรับการเขียนโปรแกรม ภาษาซี ของ Arduino ได้ทันทีมาเป็นหน่วยประมวลผลหลัก โดยชิพรุ่นนี้ มีหน่วยความจำแฟลชสำหรับเก็บเขียนโปรแกรม 128 กิโลไบต์ มีหน่วยความจำแรม 8 กิโลไบต์ มี EEPROM อีก 4 กิโลไบต์สำหรับใช้เป็นที่เก็บข้อมูลถาวรได้เมื่อไม่มีไฟเลี้ยง และยังมีขานำเข้า/ส่งออก หรือไอ/โอ (I/O) สำหรับต่อใช้งานทั่วไป 86 ขา มี PWM (สำหรับควบคุม อัตราการหมุนของมอเตอร์) ที่กำหนดความละเอียดได้ระดับ 16 บิตให้ใช้งานได้ถึง 12 ช่องสัญญาณ มีช่องสื่อสารแบบอนุกรม 4 พอร์ต และสามารถแปลงสัญญาณแอนะล็อกเป็นดิจิทัล (ADC) อีก 16 ช่องสัญญาณ ซึ่งจะเห็นว่ามี ความสามารถพื้นฐานที่มากพอสำหรับงานควบคุมที่หลากหลายเลยทีเดียว



บทความนี้เขียนขึ้นมาเพื่อแนะนำชิพ ATmega1280 ที่ใช้กับบอร์ด ET-EASY MEGA1280 โดยเนื้อหา ครอบคลุมในเรื่องของคุณสมบัติของชิพ และสถาปัตยกรรมของหน่วยประมวลผล อันได้แก่ หน้าทีของพอร์ต เรจิสเตอร์ทั่วไป เรจิสเตอร์สถานะการทำงาน และเรจิสเตอร์ชี้ตำแหน่งของสแตค ซึ่งจะได้เป็นข้อมูลพื้นฐาน ประกอบการตัดสินใจเลือกใช้ผลิตภัณฑ์ของบริษัทในชิพตระกูล ATmega

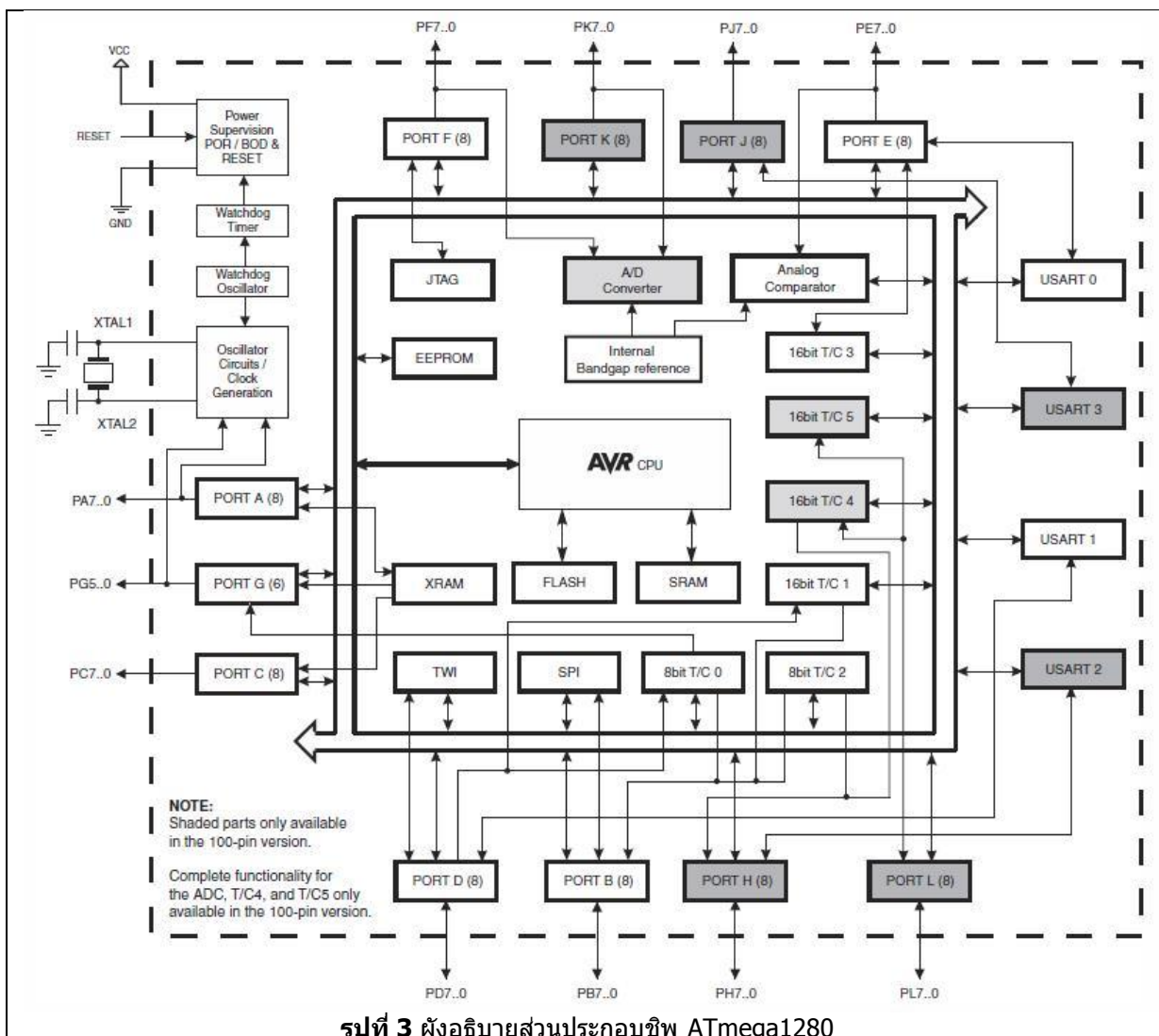
คุณสมบัติ

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ประสิทธิภาพสูงแต่ใช้พลังงานต่ำ ในตระกูล AVR®
- สถาปัตยกรรมแบบ RISC
 - มีชุดคำสั่ง 135 คำสั่ง และส่วนใหญ่คำสั่งเหล่านี้จะใช้เพียง 1 สัญญาณนาฬิกาในการประมวลผลคำสั่ง
 - มีเรจิสเตอร์สำหรับใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัว
 - ทำงานได้สูงสุดที่ 16 ล้านคำสั่งต่อวินาที (MIPS) เมื่อใช้สัญญาณนาฬิกา 16 เมกะเฮิรตซ์ (MHz)
- หน่วยความจำ
 - หน่วยความจำแฟลชสำหรับโปรแกรมขนาด 128 กิโลไบต์ เขียน/ลบได้ 10,000 ครั้ง
 - หน่วยความจำแบบ EEPROM ขนาด 4 กิโลไบต์ เขียน/ลบได้ 100,000 ครั้ง
 - หน่วยความจำแรมชนิดเอสแรม (SRAM) ขนาด 8 กิโลไบต์
 - เก็บข้อมูลได้กว่า 20 ปีที่อุณหภูมิ 85°C และกว่า 100 ปีที่อุณหภูมิ 25°C
- มีระบบโปรแกรมตัวเองอยู่ในตัวชิพ
- สามารถทำการอ่านขณะเขียนได้จริง โดยสามารถล็อกการทำงานได้เพื่อความปลอดภัยของซอฟต์แวร์
- มีการเชื่อมประสานกับ JTAG (IEEE std. 1149.1 compliant)
- คุณสมบัติการเชื่อมต่อกับอุปกรณ์ภายนอก
 - มีตัวตั้งเวลาและตัวนับขนาด 8 บิต จำนวน 2 ตัว ที่สามารถแยกโหมดการทำงานจากกันได้ 2 โหมด คือ Prescaler และ Compare
 - มีตัวตั้งเวลาและตัวนับขนาด 16 บิต จำนวน 4 ตัว ที่แยกโหมดการทำงานได้ 3 โหมด คือ Prescaler, Compare- และ Capture
 - มีตัวนับแบบเวลาจริง (Real Time Counter) ที่แยกวงจรถูกกำหนดความถี่ได้
 - มี PWM จำนวน 12 ช่องสัญญาณที่สามารถกำหนดความละเอียดได้ 16 บิต
 - มีตัวปรับผลการเปรียบเทียบของเอาต์พุต
 - มีตัวแปลงสัญญาณแอนาล็อกให้เป็นดิจิทัลขนาด 10 บิต จำนวน 16 ช่องสัญญาณ
 - มีพอร์ตสื่อสารอนุกรมที่สามารถกำหนดอัตราการรับ/ส่งได้จำนวน 4 พอร์ต
 - เชื่อมประสานอนุกรมแบบ SPI ได้ทั้งการเป็นมาสเตอร์และสเลฟ (Master/Slave)
 - มีการเชื่อมประสานแบบอนุกรมด้วยสายสัญญาณ 2 เส้นแบบ ส่งข้อมูลแบบเรียงไบต์ (Byte Oriented)
 - มีตัวตั้งเวลาแบบวอตซ์ด็อกที่สามารถกำหนดการทำงานได้โดยสามารถแยกสัญญาณนาฬิกาได้จากตัวชิพ
 - มีตัวเปรียบเทียบสัญญาณแบบแอนาล็อกอยู่ในตัว
 - มีการรองรับการขัดจังหวะและการเวก-อัพ (Wake-up) เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับขาของชิพ
- คุณสมบัติพิเศษ
 - มีระบบเริ่มระบบเมื่อมีการรีเซ็ตและมีระบบตรวจจับการเกิดบราวน์เอาต์ (Brown-out) ที่สามารถกำหนดการทำงานได้
 - มีตัวตรวจหาความเที่ยงตรงของออสซิลเลเตอร์อยู่ในตัว (Internal Calibrated Oscillator)
 - มีแหล่งการขัดจังหวะทั้งภายในและภายนอก (External and Internal Interrupt Sources)
 - มีโหมดการทำงานสลับ 6 แบบ คือ : Idle, ADC Noise Reduction, Power-save, Power-down, Standby, และ Extended Standby
- ไอ/โอ และตัวถัง
 - มีขาของไอ/โอที่สามารถกำหนดการทำงานได้ 86 ขา
 - ตัวถังแบบ TQFP ชนิด 100 ขา
- ช่วงอุณหภูมิที่ชิพทำงานได้ -40°C ถึง 85°C
- การใช้พลังงาน
 - โหมดการทำงาน: ที่ 1 MHz ต้องการแรงดัน 1.8V กระแส 500 μ A
 - โหมดเพาเวอร์ดาวน์ (Power-down) ต้องการกระแสเพียง 0.1 μ A ที่แรงดัน 1.8V



รูปที่ 2 การจัดเรียงขาของ ATmega1280

สถาปัตยกรรมของชิพ



รูปที่ 3 ผังอธิบายส่วนประกอบชิพ ATmega1280

จากรูปที่ 3 เรามาดูหน้าที่ของพอร์ต A ถึง L ว่าแต่ละพอร์ตนั้นทำหน้าที่อย่างไร

Port A (PA7..PA0)

พอร์ต A เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง (bi-directional I/O port) ขนาด 8 บิต ที่มีวงจรถวลล์พอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

Port B (PB7..PB0)

พอร์ต B เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรถวลล์พอยู่ภายใน (เลือกได้เป็นรายบิต) สามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส จุดเด่นของพอร์ตนี้ก็คือ เป็นพอร์ตที่มีความสามารถในการขับเคลื่อนได้ดีกว่าพอร์ตอื่นๆ

Port C (PC7..PC0)

พอร์ต C เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรถวลล์พอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

Port D (PD7..PD0)

พอร์ต D เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรถวลล์พอย์ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และชอร์ส

Port E (PE7..PE0)

พอร์ต E เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรถวลล์พอย์ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และชอร์ส

Port F (PF7..PF0)

พอร์ต F รองรับการนำเข้าเพื่อทำการแปลงสัญญาณแอนาล็อกมาเป็นดิจิทัล

พอร์ต F เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรถวลล์พอย์ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และชอร์ส

พอร์ต F รองรับหน้าที่การเชื่อมประสานกับ JTAG และถ้ามีการเปิดการทำงานการเชื่อมประสานกับ JTAG ตัวพวลล์พอย์ของขา PF7(TDI), PF5(TMS), และ PF4(TCK) จะทำงานจนกว่าจะเกิดการรีเซ็ต

Port G (PG5..PG0)

พอร์ต G เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรถวลล์พอย์ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และชอร์ส

Port H (PH7..PH0)

พอร์ต H เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรถวลล์พอย์ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และชอร์ส

Port J (PJ7..PJ0)

พอร์ต J เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรถวลล์พอย์ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และชอร์ส

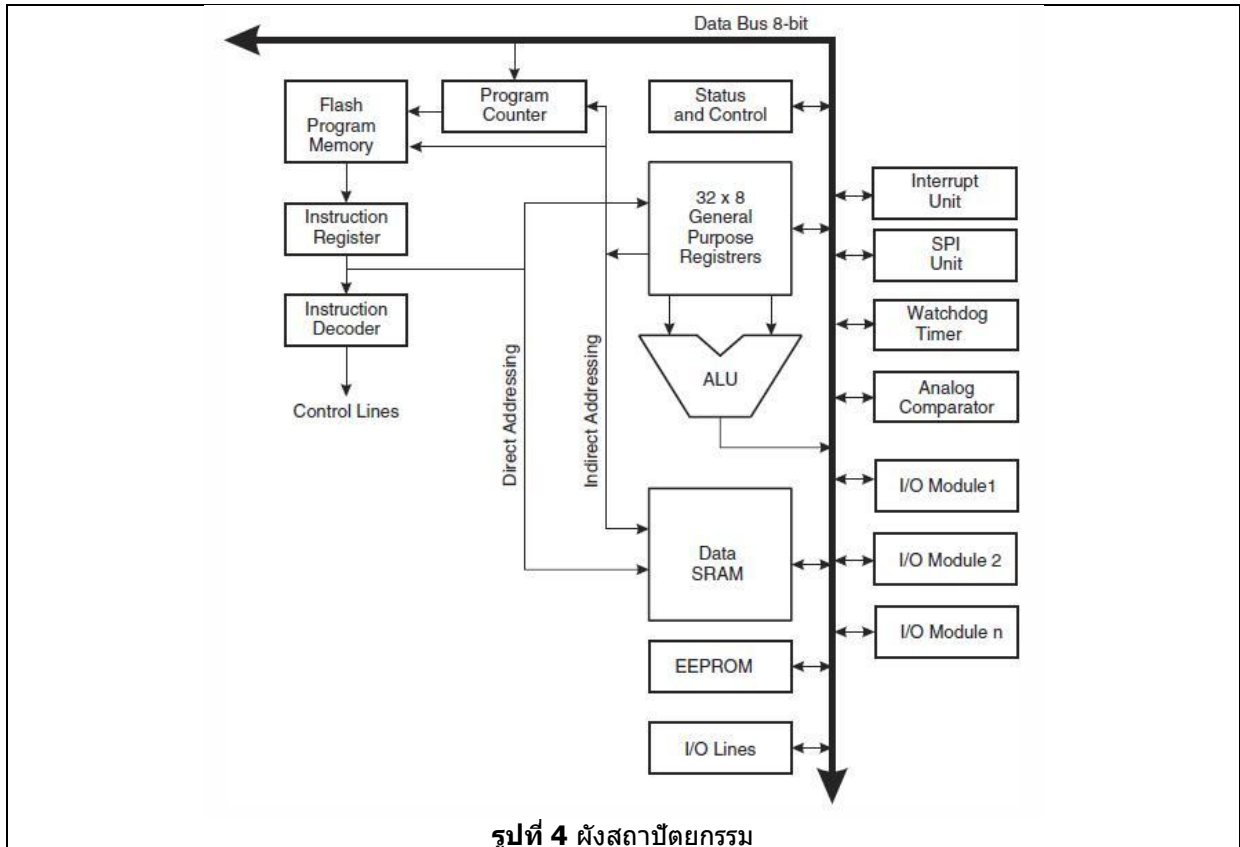
Port K (PK7..PK0)

พอร์ต K รองรับการนำเข้าเพื่อทำการแปลงสัญญาณแอนาล็อกมาเป็นดิจิทัล

พอร์ต K เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรถวลล์พอย์ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และชอร์ส

Port L (PL7..PL0)

พอร์ต L เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรถวลล์พอย์ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และชอร์ส



รูปที่ 4 ฟังก์ชันสถาปัตยกรรม

เรจิสเตอร์เก็บสถานะของ AVR: SREG

เรจิสเตอร์เก็บสถานะจัดเก็บสารสนเทศเกี่ยวกับผลของการทำงานที่เกิดจากการประมวลผลชุดคำสั่งเกี่ยวกับการคำนวณครั้งหลังสุด ซึ่งข้อมูลนี้มีประโยชน์ต่อการนำไปใช้เป็นเงื่อนไขสำหรับการทำงานของคำสั่งถัดไป

สิ่งที่จะต้องระลึกอยู่เสมอก็คือ เรจิสเตอร์ประเภทนี้จะถูกปรับปรุ่งค่าหลังจากมีการดำเนินงานด้วยหน่วยประมวลผลทางคณิตศาสตร์และตรรกะหรือเอแอลยู (ALU: Arithmetic and Logic Unit) ซึ่งเราสามารถหารายละเอียดของผลที่เกิดขึ้นได้จากเอกสารของชิพ

เรจิสเตอร์เก็บสถานะจะไม่ ถูกจัดเก็บโดยอัตโนมัติเมื่อเข้าสู่การเรียกโปรแกรมย่อยที่ดอบสนองการขัดจังหวะและไม่ถูกนำกลับคืนหลังจากที่การขัดจังหวะนั้นเสร็จสิ้นแล้ว นั่นหมายความว่า ผู้เขียนโปรแกรมจะต้องดำเนินการสิ่งเหล่านี้ด้วยตนเอง

จากรูปที่ 5 จะเห็นว่ามีบิตเก็บสถานะของการทำงานด้วยกัน 8 บิต คือ

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

รูปที่ 5 โครงสร้าง SREG

• Bit 7 – I: Global Interrupt Enable

บิตการเปิดการทำงานของการขัดจังหวะแบบโกลบอลจะถูกกำหนดเป็น 1 หรือถูกเซต (set) เพื่อยอมให้มีการทำการขัดจังหวะได้ แต่ถ้าบิตนี้เป็น 0 หรือถูกล้างค่า (cleared) จะหมายความว่าไม่มีการเกิดการขัดจังหวะ ณ ขณะเวลานั้น นั่นหมายความว่า บิต I จะถูกล้างค่าโดยฮาร์ดแวร์เองเมื่อมีการขัดจังหวะไปเป็นที่เรียบร้อยแล้ว และจะถูกเซตค่าด้วยคำสั่ง RETI เพื่อทำการขัดจังหวะครั้งต่อไป นอกจากนี้เราสามารถเซตและล้างค่าบิต I ได้ด้วยคำสั่ง SEI และ CLI

- **Bit 6 – T: Bit Copy Storage**

คำสั่งคัดลอกบิตที่ชื่อว่า BLD (Bit Load) และ BST (Bit Store) จะใช้บิต T เป็นแหล่งข้อมูลบิตหรือที่เก็บบิตระหว่างที่ทำงาน นั้นหมายความว่า เมื่อเราสั่ง BST จะมีการนำบิตจากเรจิสเตอร์มาเก็บเอาไว้ที่บิต T และเมื่อเราใช้คำสั่ง BLD ก็จะทำบิตจากบิต T ไปเก็บในเรจิสเตอร์

- **Bit 5 – H: Half Carry Flag**

บิต H จะใช้กับการคำนวณเกี่ยวกับ BCD

- **Bit 4 – S: Sign Bit**

บิต S เป็นที่เก็บผลของเครื่องหมายของตัวเลข โดยคำนวณมาจาก $S = N \oplus V$

- **Bit 3 – V: Two's Complement Overflow Flag**

บิต V ใช้ในการสนับสนุนการทำ 2 คอมพลีเมนต์ (Two's complement) ของตัวเลข

- **Bit 2 – N: Negative Flag**

บิต N เป็นตัวบ่งชี้ว่าจากการคำนวณหรือดำเนินการทางตรรกศาสตร์นั้นก่อให้เกิดผลลัพธ์ค่าลบหรือไม่

- **Bit 1 – Z: Zero Flag**

บิต Z เป็นตัวชี้ให้รู้ว่าจากการคำนวณหรือดำเนินการทางตรรกศาสตร์แล้วเกิดค่าเป็น 0 หรือไม่

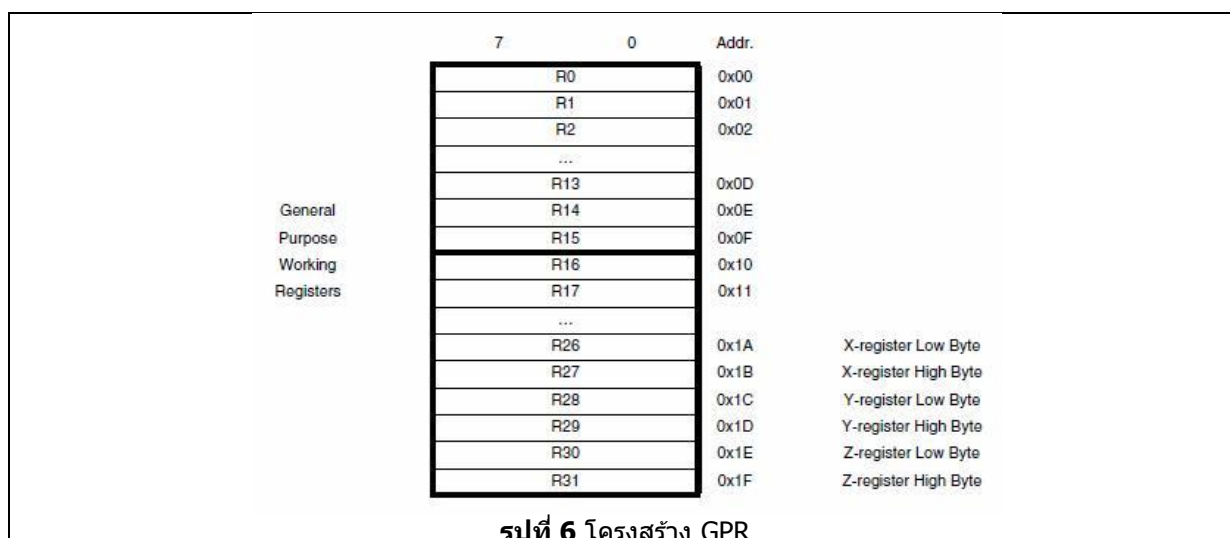
- **Bit 0 – C: Carry Flag**

บิต C เป็นที่เก็บผลว่าจากการคำนวณหรือดำเนินการทางตรรกศาสตร์แล้วเกิดการยืมบิตกันหรือไม่

เรจิสเตอร์ใช้งานทั่วไป

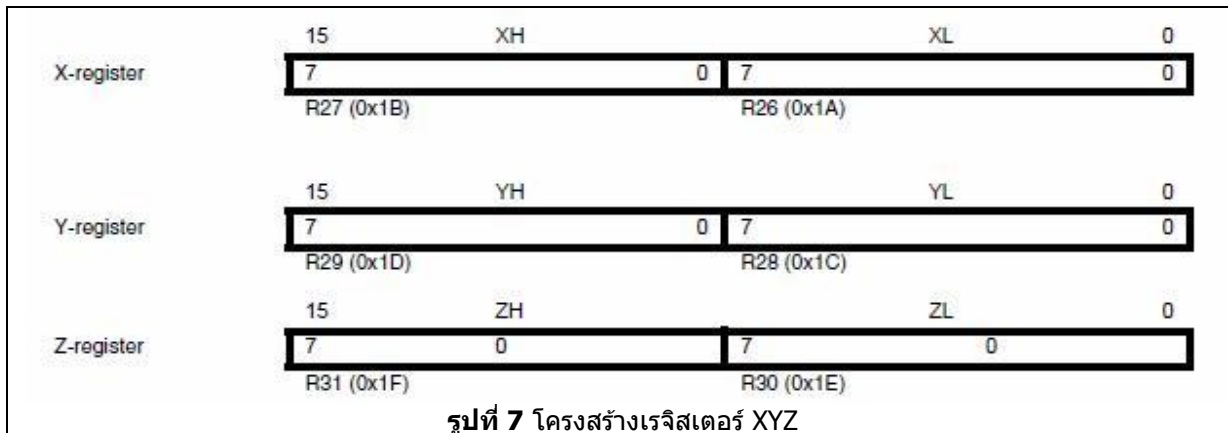
เรจิสเตอร์ใช้งานทั่วไปหรือจีพีอาร์ (GPR File : General Purpose Register File) เป็นเรจิสเตอร์ที่ถูกกำหนดขึ้นมาเพื่อใช้กับชุดคำสั่งแบบ RISC ของ AVR ซึ่งในการทำงานของคำสั่งนั้นจะมีการนำเข้าข้อมูล (output operand) จากเรจิสเตอร์ไปใช้งาน และส่งเข้าข้อมูล (input operand) จากคำสั่งมาเก็บในเรจิสเตอร์ดังนี้

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input



เรจิสเตอร์ X, Y และ Z

จากรูปที่ 6 จะมีการมองก่อนของเรจิสเตอร์ให้เป็น 16 บิต จำนวน 3 ตัว คือ เรจิสเตอร์ X, Y และ Z ซึ่งแต่ละตัวจะใช้เรจิสเตอร์ตามรูปที่ 7



ตัวชี้สแตก

สแตกถูกใช้งานสำหรับการเก็บข้อมูลชั่วคราว เช่น ตัวแปรภายใน (local variables) เก็บตำแหน่งของตำแหน่งถัดไปที่จะต้องกระทำก่อนมีการขัดจังหวะหรือเรียกโปรแกรมย่อย และตัวชี้ตำแหน่งสแตก จะมีหน้าที่เก็บค่าตำแหน่งสูงสุดของสแตก (TOS: Top Of Stack) เอาไว้

ตัวชี้สแตกจะชี้ไปยังข้อมูลในหน่วยความจำแรมที่ถูกใช้เป็นสแตก ตามแต่ที่โปรแกรมย่อยหรือโปรแกรมตอบสนองการขัดจังหวะกำหนดเอาไว้

ในชิพ AVR นั้นตัวชี้สแตกจะเป็นเรจิสเตอร์ขนาด 8 บิต จำนวน 2 ตัว เรียกว่า SPH และ SPL (ดังรูปที่ 8)

Bit	15	14	13	12	11	10	9	8	
0x3E (0x5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
0x3D (0x5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	1	
	1	1	1	1	1	1	1	1	

รูปที่ 8 โครงสร้างตัวชี้สแตก

สรุป

จากคุณสมบัติของชิพ ATmega 1280 รวมถึงหน้าที่ของพอร์ตและเรจิสเตอร์ที่เกี่ยวข้อง คงเพียงพอสำหรับเป็นข้อมูลเบื้องต้นสำหรับเป็นจุดเริ่มต้นสำหรับผู้สนใจศึกษาการเขียนโปรแกรมด้วยชิพดังกล่าว ดังนั้น ถ้าต้องการทราบรายละเอียดที่มากกว่านี้ ผู้เขียนแนะนำให้ท่านอ่านได้จากเอกสารประกอบชิพที่ทางบริษัท Atmel ได้เตรียมไว้ให้แล้ว หรือดาวน์โหลดเอกสารได้จากบริษัทที่ที่ ครับ

สุดท้าย ขอขอบคุณทางบริษัทที่ที่ที่เปิดโอกาสให้ผมได้เขียนบทความด้านนี้อีกครั้งหลังจากที่ห่างเหินกันไปนานเลยทีเดียว และหวังว่าคงพอเป็นประโยชน์สำหรับผู้สนใจไม่มากนักน้อยครับผม