บทที่ 4
ไอเอสเอและสตับ

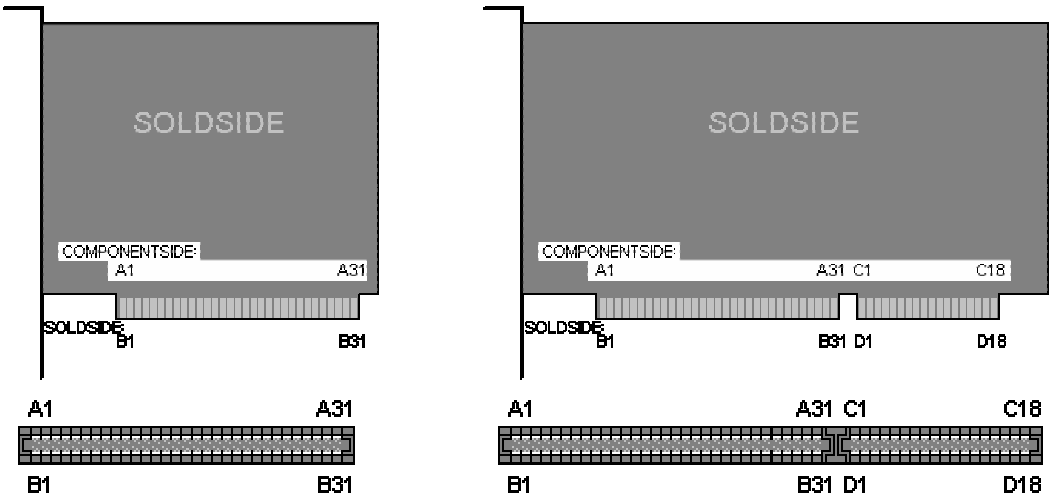
อุปกรณ์ที่เป็นฮาร์ดแวร์ในโครงงานนี้ที่ใช้ทำการเชื่อมต่อกับเครื่องคอมพิวเตอร์นั้น จะเชื่อมต่อกันผ่านทางสล็อต (Slot) ของไอเอสเอ (ISA) โดยผ่านทางไอเอสเอบัส (ISA Bus หรือ เรียกว่า ไอซ่าบัส) เพราะไอเอสเอถือเป็นพื้นฐานของการเชื่อมต่อข้อมูลภายใน (Internal Bus) เครื่องคอมพิวเตอร์ ซึ่งในส่วนของโปรแกรมการเข้ารหัสและในส่วนของไอเอสเอบัส จะมีปัญหาในการเชื่อมต่อการทำงานร่วมกัน จึงได้ทำการสร้างในส่วนของสตับ (Stub) ขึ้น เพื่อให้ทั้งสองส่วนสามารถทำงานร่วมกันได้

4.1 การ์ดไอเอสเอ (ISA Card)

หมายถึง การ์ดที่ทำการเชื่อมต่อกับเครื่องคอมพิวเตอร์โดยผ่านทางไอเอสเอบัส ซึ่งคำว่าไอเอสเอย่อมาจาก Industry Standard Architecture

4.1.1 ไอเอสเอบัส

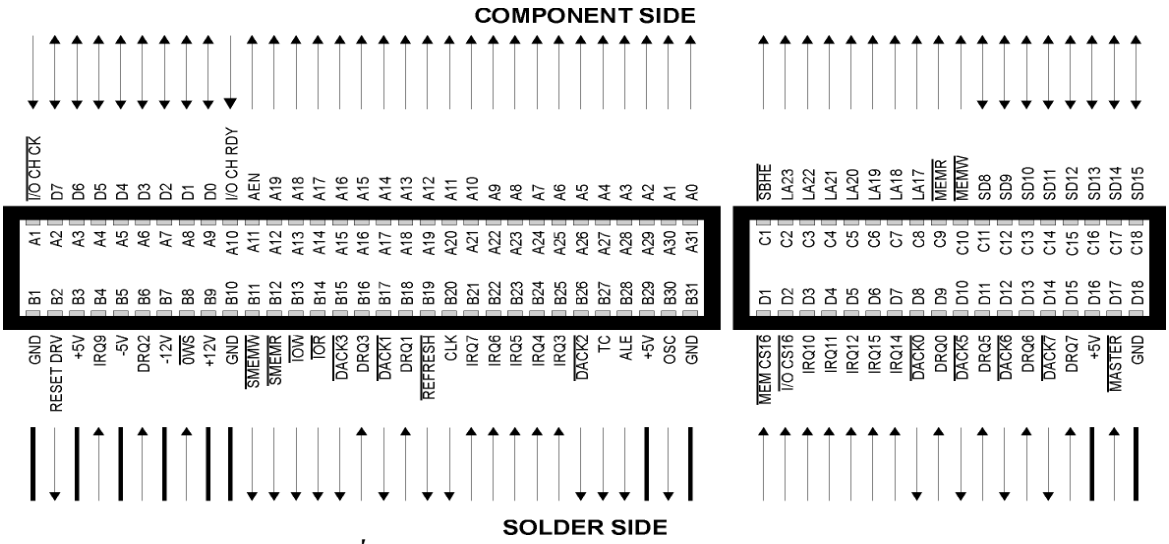
ไอเอสเอบัส หรือเรียกอีกชื่อหนึ่งว่า เอทีบัส (AT Bus) เป็นระบบขนถ่ายข้อมูลอุปกรณ์ภายใน ซึ่งไม่ต้องการความเร็วสูงมากนัก ซึ่งทั้งพีซีไอ (PCI) ไอเอสเอ หรือ เอจีพี (AGP) ล้วนหมายถึง การ์ดเพิ่มขยาย (Expansion Card) ที่นำมาต่อกับบัสเพิ่มขยาย (Expansion Bus) เพื่อช่วยเพิ่มประสิทธิภาพและเพิ่มขีดความสามารถให้กับเครื่องคอมพิวเตอร์ เนื่องจากระบบบัสเพิ่มขยายสามารถปรับแต่งหรือเพิ่มขยายความสามารถของระบบได้ โดยผ่านทางปลั๊กอินบอร์ด (Plug-in Board) หรือเรียกว่าบัสเพิ่มขยาย เช่น เมื่ออยากให้คอมพิวเตอร์เล่นเพลงได้ ก็ต้องซื้อซาวนด์การ์ด (Soundcard) และลำโพง โดยแค่นำมาปลั๊กลงในสล็อตเพิ่มขยาย (Expansion Slot) บนเมนบอร์ด (Mainboard) และทำการติดตั้งก็สามารถใช้งานได้ทันทีโดยไม่จำเป็นต้องรื้อเมนบอร์ดใหม่



รูปที่ 4-1: ลักษณะของการ์ดและสล็อตของการ์ด 8 บิต และ 16 บิต

การ์ดไอเอสเอนั้นมีอยู่ 2 ขนาด คือ การ์ดขนาด 8 บิต (Bit) และ 16 บิต ซึ่งเราสามารถนำการ์ดขนาด 8 บิตมาเสียบบนช่อง 16 บิตได้ เพราะใช้สถาปัตยกรรมพื้นฐานเหมือนกัน จะต่างกันตรงส่วนที่เพิ่มมาสำหรับ 16 บิตเท่านั้น

4.1.2 ขาสัญญาณต่าง ๆ ของการ์ดไอเอสเอ



รูปที่ 4-2: ขาสัญญาณต่าง ๆ ของการ์ดไอเอสเอ

จากรูปนั้น จะเห็นถึงขาสัญญาณ (Signal) ต่าง ๆ ของการ์ดไอเอสเอ ซึ่งจะแบ่งออกเป็น 2 ส่วน คือ

1. ส่วนคอมโพเน็นต์ (Component Side)
2. ส่วนโซลเดอร์ (Solder Side)

0WS

Zero Wait State จะถูกเป็นเซตเป็น low โดยอุปกรณ์รองที่มีการใช้บัส (Bus Slave Device) เพื่อบอกว่า มันสามารถทำงานได้โดยที่ไม่ต้องมีเพิ่มสถานะ wait (Wait state) ซึ่งจะทำงานที่ไซเกิล (Cycle) ของหน่วยความจำเป็น 16 บิต โดยขาสัญญาณ –0WS ถูก derive มาจากการ decode แอดเดรส

AEN

Address Enable ถูกใช้เพื่อ degate ระบบไมโครโพรเซสเซอร์ (Microprocessor) และอุปกรณ์อื่น ๆ ผ่านบัสในระหว่างที่ DMA ส่งข้อมูล และเมื่อสัญญาณนี้ active ระบบตัวควบคุม DMA (System DMA Controller) จะสามารถควบคุมขา Address ขา Data และควบคุมขา Read และ Write ได้ ซึ่งควรที่จะรวมขาสัญญาณนี้ไปกับการเลือกบอร์ด เพื่อป้องกันการเลือกบอร์ดผิดระหว่างที่ DMA ทำงาน

BALE

Buffered Address Latch Enable ถูกใช้เพื่อ latch ขาสัญญาณ LA23 ถึง LA17 หรือ decode ของขาสัญญาณเหล่านี้ โดยขา Address จะถูก latch ที่ขอบขาลงของขาสัญญาณ BALE ซึ่งมันจะถูกเซตเป็น high ตลอดการทำงานของ DMA ซึ่งเมื่อนำมาใช้กับขาสัญญาณ AEN แล้ว มันจะบ่งบอกตัวไมโครโพรเซสเซอร์หรือแอดเดรสของ DMA

CLK

System Clock เป็น คล็อก (Clock) ที่วิ่งโดยอิสระอยู่ในช่วง 8 ถึง 10 เมกะเฮิรตซ์ ซึ่งมันจะถูกใช้กับบางแอพพลิเคชันของบอร์ดไอเอสเอที่อนุญาตให้ทำงานเป็นซิงโครนัส (Synchronous) โดยระบบไมโครโพรเซสเซอร์

DACK0 to DACK3 and DACK5 to DACK7

DMA Acknowledge 0 ถึง 3 และ 5 ถึง 7 ถูกใช้เป็นสัญญาณ acknowledge ตอบรับการร้องขอขาสัญญาณ DRQ0 ถึง DRQ3 และ DRQ5 ถึง DRQ7 ของ DMA

DRQ0 to DRQ3 and DRQ5 to DRQ7

DMA Request ถูกใช้โดยบอร์ดไอเอสเอเพื่อทำการร้องขอบริการ (Request Service) จากระบบตัวควบคุม DMA หรือทำการร้องขอการเป็นเจ้าของบัสเพื่อนำบัสมาใช้งาน (สัญญาณเหล่านี้อาจจะถูกเซตให้ทำงานเป็นแบบอะซิงโครนัส (Asynchronous)) โดยอุปกรณ์ที่มาทำการร้องขอนั้น จะถูกเซตขาสัญญาณการร้องรอเป็น active ไปจนกว่าไอเอสเอบอร์ดจะตอบรับสัญญาณ DACK

I/O CS16

I/O Chip Select 16 จะถูกเซตเป็น low โดยอุปกรณ์รองที่มีการติดต่อกับอินพุต-เอาต์พุต (I/O slave device) เพื่อบอกว่า มันสามารถทำการขนถ่ายข้อมูลที่เป็นอินพุต-เอาต์พุต 16 บิตได้ ซึ่งสัญญาณนี้ถูกเซตจากการ decode แอดเดรส SA15 ถึง SA0

I/O CH CK

สัญญาณ I/O Channel Check อาจจะถูกกระตุ้น (โดยการร้องขอของบอร์ดไอเอสเอ) มากกว่าสัญญาณเอ็นเอ็มไอ (NMI: Non-Maskable Interrupt) ซึ่งถูก generate โดยระบบไมโครโพรเซสเซอร์ ซึ่งสัญญาณจะถูกเซตเป็น active เพื่อบอกถึงความผิดพลาดที่เกิดขึ้นเพื่อให้ทำการตรวจสอบ

I/O CH RDY

ในขณะที่มีการใช้ไอโอหรือหน่วยความจำอยู่ I/O Channel Ready จะเซตให้ตัวที่ต้องการจะใช้ไอโอหรือใช้หน่วยความจำเช่นกัน แต่ว่ามาทีหลังให้รอก่อนโดยการแทรกสถานะ wait เข้าไป ขาสัญญาณนี้จะถูกเซตเป็น high อยู่เสมอ (หมายถึงพร้อมทำงาน) ซึ่งบอร์ดไอเอสเอจะเซตขาสัญญาณนี้เป็น low เมื่อไม่พร้อมทำงานหรือทำงานไม่เสร็จและทำการเพิ่มสถานะ wait เข้าไป โดยตัวอุปกรณ์ที่มีการใช้ขาสัญญาณนี้ที่มีการแทรกสถานะ wait ควรที่จะเซตขาสัญญาณนี้เป็น low ทันทีที่ทำการตรวจพบว่ามีการใช้สัญญาณขา Address และมีการทำคำสั่งอ่านและ

เขียนอยู่ และจะเซตกลับมาเป็น high ใหม่เมื่ออุปกรณ์ทำไซเกิลของไอโอหรือหน่วยความจำที่เพิ่มสถานะ wait เสร็จแล้ว

IOR

I/O Read จะถูกสั่งงานจากเจ้าของบัสขณะนั้น โดยจะสั่งให้อุปกรณ์อินพุต-เอาต์พุตที่ถูกเลือกทำการอ่านข้อมูลจาก Data bus

IOW

I/O Read จะถูกสั่งงานจากเจ้าของบัสขณะนั้น โดยจะสั่งให้อุปกรณ์อินพุต-เอาต์พุตที่ถูกเลือกทำการเขียนข้อมูลลงบน Data bus

IRQ3 to IRQ7 and IRQ9 to IRQ12 and IRQ14 to IRQ15

Interrupt Request จะถูกใช้เป็นตัวให้สัญญาณแก่ตัวไมโครโพรเซสเซอร์ เมื่อบอร์ดไอเอสเอต้องการที่จะเรียกใช้งาน ซึ่งสัญญาณ IRQ จะทำการเปลี่ยนจาก low เป็น high จนกว่าไมโครโพรเซสเซอร์จะมีสัญญาณ ACK ตอบกลับการร้องขอ interrupt service routine กลับมา โดยจะใช้ความสำคัญของ IRQ9 ถึง IRQ12 และ IRQ14 ถึง IRQ15 เป็น highest priority และ IRQ3 ถึง IRQ7 เป็น lowest priority

LA23 to LA17

Unlatched Address bits 23:17 ใช้เป็นแอดเดรสเพื่อเข้าถึงหน่วยความจำภายในระบบ ซึ่งจะใช้งานร่วมกับขาสัญญาณ SA19 ถึง SA0 เพื่อให้สามารถอ้างอิงหน่วยความจำได้ 16 เมกะไบต์ โดยจะทำงานเมื่อขาสัญญาณ BALE เป็น high ซึ่งมันจะไม่ถูก latch และไม่นำมาใช้ในบัสไซเกิล โดย decode ของขาสัญญาณเหล่านี้ ควรที่จะถูก latch ที่ขอบขาลงของขาสัญญาณ BALE

MASTER

Master ถูกใช้โดยบอร์ดไอเอสเอร่วมกับขาสัญญาณ DRQ เพื่อขอเป็นเจ้าของบัสเพื่อนำบัสมาใช้งาน โดยเมื่ออุปกรณ์รับสัญญาณ –DACK มา อุปกรณ์จะทำการเซตสัญญาณ –MASTER ให้เป็น low ซึ่งจะเป็นการอนุญาตให้ควบคุมขา Address ขา Data และขาควบคุมต่าง ๆ และหลังจากที่เซตสัญญาณ –MASTER ให้เป็น low แล้ว อุปกรณ์ควรที่จะรออีก 1 ช่วงคล็อก (Clock Period) ก่อนที่จะทำการใช้ขา Address และขา Data และรออีก 2 ช่วงคล็อกก่อนที่จะทำคำสั่งอ่านหรือเขียน

MEM CS16

Memory Chip Select 16 จะถูกเซตเป็น low โดยอุปกรณ์รองที่มีหน่วยความจำ (Memory Slave Device) เพื่อบอกว่า มันสามารถทำการขนถ่ายข้อมูลที่เป็นหน่วยความจำ 16 บิตได้ ซึ่งสัญญาณนี้ถูกเซตจากการ decode แอดเดรส LA23 ถึง LA17

MEMR

Memory Read ใช้ในการเลือกอุปกรณ์ที่มีหน่วยความจำ (Memory Device) ซึ่งจะเป็นการนำข้อมูลลงใน Data bus โดยขาสัญญาณจะ active อยู่ตลอดการทำงานของไซเกิลของการอ่านหน่วยความจำ (Memory Read Cycle)

MEMW

Memory Write ใช้ในการเลือกอุปกรณ์ที่มีหน่วยความจำ ซึ่งจะเป็นการนำข้อมูลขึ้นมาจาก Data bus โดยขาสัญญาณจะ active อยู่ตลอดการทำงานของไซเกิลของการเขียนหน่วยความจำ (Memory Write Cycle)

OSC

Oscillator เป็นคล็อกที่มีช่วงเวลา 70 นาโนเซ็ก (ns) ซึ่งสัญญาณนี้ไม่ซิงโครนัสกับคล็อกของระบบ (System Clock: CLK)

REFRESH

Memory Refresh จะถูกเซตเป็น low เพื่อบอกให้ทำการ refresh หน่วยความจำ

RESET DRV

Reset Drive จะถูกเซตเป็น high เพื่อทำการ reset หรือ ตั้งค่าโลจิกใหม่ (Initialize) เมื่อทำการเปิดเครื่องขึ้นมาใหม่ หรือมีการกด reset

SA19 to SA0

System Address bits 19:0 ใช้เป็นแอดเดรสที่เข้าถึงหน่วยความจำและอุปกรณ์อินพุต-เอาต์พุตภายในระบบ ซึ่งสัญญาณเหล่านี้อาจจะนำมาใช้ร่วมกับขาสัญญาณ LA23 ถึง LA17 เพื่อให้สามารถอ้างแอดเดรสของหน่วยความจำได้มากขึ้นถึง 16 เมกะไบต์ แต่มีเพียง 16 บิตล่างเท่านั้นที่ใช้เกี่ยวข้องกับตัวอินพุตและเอาต์พุต ซึ่งจะใช้อ้างแอดเดรสได้ 64K

ขา SA19 เป็นบิตที่มีสำคัญมากที่สุด (The Most Significant Bit) และขา SA0 เป็นบิตที่มีความสำคัญน้อยที่สุด (The Least Significant Bit) โดยสัญญาณเหล่านี้จะถูกนำเข้าสู่ระบบบัสเมื่อขาสัญญาณ BALE เป็น high และจะถูก latch ที่ขอบขาลงของขาสัญญาณ BALE ส่วนสัญญาณที่เหลือจะใช้สำหรับการ read และ write ซึ่งสัญญาณทั้งหมดจะถูกควบคุมโดยระบบไมโครโพรเซสเซอร์หรือ DMA หรือไม่ก็อาจจะโดยเจ้าของบัสขณะนั้นบนไอเอสเอบอร์ด

SBHE

System Byte High Enable จะถูกเซตเป็น low เพื่อบอกถึงการขนถ่ายข้อมูลบน Data bus D15 ถึง D8

SD15 to SD0

System Data เปรียบเสมือนเป็น Data bus สำหรับอุปกรณ์บนไอเอสเอบัส ซึ่งขา SD15 เป็นบิตที่มีความสำคัญมากที่สุด และขา SD0 เป็นบิตที่มีความสำคัญน้อยที่สุด โดยที่ขาสัญญาณ SD7 ถึง SD0 ใช้ในการขนถ่ายข้อมูลของอุปกรณ์ที่มีขนาด 8 บิต ส่วนขาสัญญาณ SD15 ถึง SD0 จะใช้ในการขนถ่ายข้อมูลของอุปกรณ์ที่มีขนาด 16 บิต หากต้องการขนถ่ายข้อมูลของอุปกรณ์ที่มีขนาด 16 บิตโดยใช้ขาสัญญาณ SD7 ถึง SD0 ก็จะใช้ 2 ไซเกิล ซึ่งไซเกิลละ 8 บิต

SMEMR

System Memory Read ใช้ในการเลือกอุปกรณ์ที่ใช้หน่วยความจำ ซึ่งจะเป็นการนำข้อมูลลงใน Data bus โดยขาสัญญาณจะ active เฉพาะตอนที่การ decode หน่วยความจำอยู่ภายใน

พื้นที่ของหน่วยความจำที่มีค่าต่ำกว่า 1 เมกะไบต์ ขาสัญญาณ SMEMR ถูก derive มาจาก MEMR และการ decode หน่วยความจำที่ค่าต่ำกว่า 1 เมกะไบต์

SMEMW

System Memory Write ใช้ในการเลือกอุปกรณ์ที่ใช้หน่วยความจำ ซึ่งเป็นการนำข้อมูลมาจาก Data bus โดยขาสัญญาณจะ active เฉพาะตอนที่การ decode หน่วยความจำอยู่ภายในพื้นที่ของหน่วยความจำที่มีค่าต่ำกว่า 1 เมกะไบต์ ขาสัญญาณ SMEMW ถูก derive มาจาก MEMW และการ decode หน่วยความจำที่ค่าต่ำกว่า 1 เมกะไบต์

TC

Terminal Count เป็นตัวสร้าง pulse สัญญาณ โดย terminal count จะถูกส่งไปยัง DMA

4.1.3 ดีเอ็มเอ (DMA: Direct Memory Access)

การรับส่งข้อมูลกับหน่วยความจำโดยตรง โดยไม่มีการส่งข้อมูลผ่านทางซีพียู (CPU) เลย ซึ่งเป็นผลให้การรับส่งข้อมูลเร็วขึ้นและทำให้ซีพียูสามารถทำงานโปรเซสอื่น ๆ ต่อไปได้ ในการรับส่งข้อมูลแบบ DMA นั้น จำเป็นต้องมีการใช้แชนแนล หรือตัวควบคุม DMA (DMA Controller) โดยแชนแนลจะทำหน้าที่แทนซีพียู คือ เมื่อต้องการจะรับส่งข้อมูลแบบ DMA ตัวแชนแนลจะทำการส่งสัญญาณไปบอกให้ซีพียูรับทราบ ซึ่งซีพียูจะสั่งให้แชนแนลทำรูทีน (Routine) ควบคุมการส่งข้อมูล จากนั้นซีพียูจะไปทำงานอื่น และเมื่อการทำ DMA เสร็จแล้ว แชนแนลจะทำการส่งสัญญาณไปบอกให้ซีพียูรับทราบอีกครั้งว่า การทำ DMA เสร็จสิ้นแล้ว

ในขณะที่ซีพียูกำลังทำงานอื่นอยู่ จะไม่เกิดการทำ DMA ขึ้น เนื่องจากแชนแนลจำเป็นต้องใช้ดาต้าบัส (Data Bus) และแอดเดรสบัส (Address Bus) ของหน่วยความจำ และซีพียูเองก็จำเป็นต้องใช้บัสทั้งสองนี้เช่นกัน ดังนั้น DMA จะเกิดขึ้นเฉพาะในช่วงที่ซีพียูไม่ต้องการใช้หน่วยความจำ (ดาต้าบัสมีแอดเดรสว่างอยู่) เท่านั้น เช่น ซีพียูกำลังคำนวณฟังก์ชันทางคณิตศาสตร์อยู่ ซึ่งไม่มีความจำเป็นต้องใช้หน่วยความจำในช่วงเวลานี้ ดังนั้นดาต้าบัสและแอดเดรสบัสจึงว่างอยู่ ทำให้เกิดการทำ DMA ขึ้น ถ้าเมื่อใดซีพียูต้องการใช้หน่วยความจำ DMA จะต้องหยุดการทำงานลงชั่วคราว จนกว่าซีพียูจะใช้งานหน่วยความจำเสร็จ DMA ถึงจะสามารถทำงานต่อได้ ช่วงเวลาขณะที่ DMA ใช้บัสทั้งสอง ในขณะที่ซีพียูไม่ได้ใช้งาน เรียกว่า “การขโมยรอบเวลา (Cycle Stealing) ”

<u>ชนิดของการใช้ขาสัญญาณของระบบบอร์ดไอเอสเอ (ISA System Board)</u>

Signal Name	System Board Usage	Signal Name	System Board Usage
AEN	Output	**MEM CS16**	I/O
BALE	Output	**MEMR**	I/O
CLK	Output	**MEMW**	I/O

DACK	Output	**OSC**	Output
DRQ	Input	**REFRESH**	I/O
I/O CS16	Input	**RESET DEV**	Output
I/O CH CK	Input	**SA**	I/O
I/O CH RDY	I/O	**SD**	I/O
IOR	I/O	**SBHE**	I/O
IOW	I/O	**SMEMR**	I/O
IRQ	Input	**SMEMW**	I/O
LA	I/O	**TC**	I/O
MASTER	Input	**0WS**	Input

ตารางที่ 4-1: ชนิดของการใช้ขาสัญญาณ โดยระบบบอร์ดไอเอสเอ (ISA System Board)

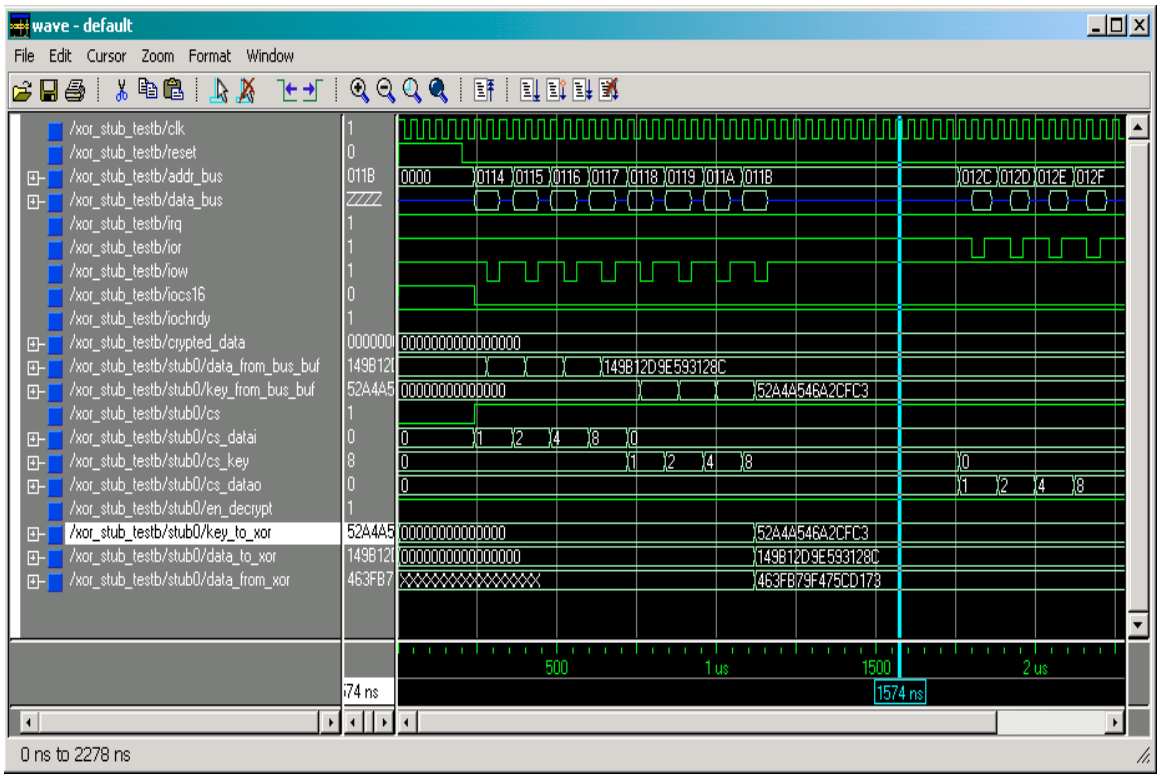
4.2 สตับ (Stub)

เป็นส่วนที่ทำหน้าที่เป็นตัวเชื่อมระหว่างกลางของสองฝั่ง เพื่อให้สามารถนำมาทำงานร่วมกันได้ โดยในขั้นตอนของการออกแบบสตับ เราได้แบ่งส่วนติดต่อออกเป็น 2 ฝั่งคือ

1) ฝั่งติดต่อกับส่วนของโปรแกรมการเข้ารหัส ในส่วนนี้จะทำการติดต่อกันภายในโปรแกรมภาษาวีเอชดีแอล (VHDL) เอง
2) ฝั่งติดต่อไอเอสเอบัสฝั่งนี้จะมีการเชื่อมต่อผ่านอุปกรณ์ทางกายภาพ(Physically) แต่เราจะเชื่อมต่อผ่านบัฟเฟอร์ (buffer) อีกทีนึง

สำหรับฝั่งติดต่อไอเอสเอบัส เราจะไม่เชื่อมต่อกับทุกเส้นแต่จะเลือกเฉพาะข้อมูลที่สำคัญต่อการเชื่อมต่อข้อมูล ซึ่งมีเส้นต่างๆดังนี้ Address(11:0), Data(15-0), I/O CHRDY, I/O CHCK, IOW, IOR, RESET DRV, IRQ, I/O CS16, AEN และอาจจะมีการเชื่อมต่อ เส้นข้อมูลอื่นๆ เท่าที่จำเป็นต่อไปในอนาคต

นอกจากส่วนติดต่อแล้ว ส่วนที่สำคัญที่สุดของสตับ คือส่วนของบัฟเฟอร์ เพราะ สตับต้องควบคุมการทำงานระหว่างโปรแกรมการเข้ารหัสจากบอร์ดเอฟพีจีเอกับไอเอสเอบัส ที่มีความกว้างของข้อมูลไม่เท่ากัน ให้สามารถทำงานร่วมกันได้ด้วยดี ซึ่งข้อมูลที่สำคัญทั้ง 2 ฝั่งจะถูกเก็บไว้นานเท่าที่จำเป็น การติดต่อข้อมูลจากไอเอสเอบัสไปยังโปรแกรมการเข้ารหัส จะเป็นการติดต่อข้อมูลระหว่างบัฟเฟอร์ของทั้ง 2 ฝั่ง และตัวบัฟเฟอร์ถึงจะส่งข้อมูลไปยังส่วนติดต่อของตัวเองด้วย ช่วงเวลาที่สอดคล้องกับขนาดความกว้างของส่วนติดต่อเอง



รูปที่ 4-3: ไดอะแกรมช่วงเวลาของสัญญาณที่สำคัญของสตับ

สัญญาณด้านที่ติดกับส่วนของบัส

clk, reset, addr_bus[], data_bus[], ior, iow, iocs16 และ iochrdy เป็นสัญญาณด้านที่ติดกับบัส ซึ่งความกว้างของขาสัญญาณจะเท่ากับความกว้างที่ใช้ในบัสนั้น ๆ ซึ่งในกรณีนี้ data_bus[] จะกว้างเป็น 16 บิต และ addr_bus[] จะกว้างเป็น 12 บิต และช่วงเวลาของสัญญาณจะขึ้นอยู่กับชนิดของบัสเช่นกัน สำหรับในกรณีของเรา จะเลือกใช้เฉพาะสัญญาณพื้นฐานที่จำเป็นของไอเอสเอบัส และการทำงานของช่วงสัญญาณจะเป็นไปตามมาตรฐานของไอเอสเอบัส ดังที่ได้กล่าวไว้ก่อนแล้ว

สัญญาณด้านที่ติดกับส่วนของการเข้ารหัส

data_from_bus_buf[], key_from_bus_buf[], cs_datai[], cs_key[] และ cs_data_out[] เป็นสัญญาณด้านที่ติดกับส่วนเข้ารหัส ซึ่งสัญญาณจะมีความกว้าง 64 บิต เท่ากับขนาดของบล็อกของการเข้ารหัสแบบ DES โดยช่วงของสัญญาณจะเป็นการส่งและรับข้อมูลขนาด 64 บิต และมีสัญญาณที่ใช้ในการบอกฝั่งปลายทางว่าข้อมูลนั้นเป็นข้อมูลที่ถูกต้องหรือไม่ เช่นสัญญาณ data_from_bus_buf[] ซึ่งจะใช้คู่กับสัญญาณ data_from_bus_buf_valid ถ้าสัญญาณ data_from_bus_buf_valid เป็น 1 แล้ว ฝั่งของส่วนการเข้ารหัส จะทำการบัฟเฟอร์ plain data อันนี้ไว้