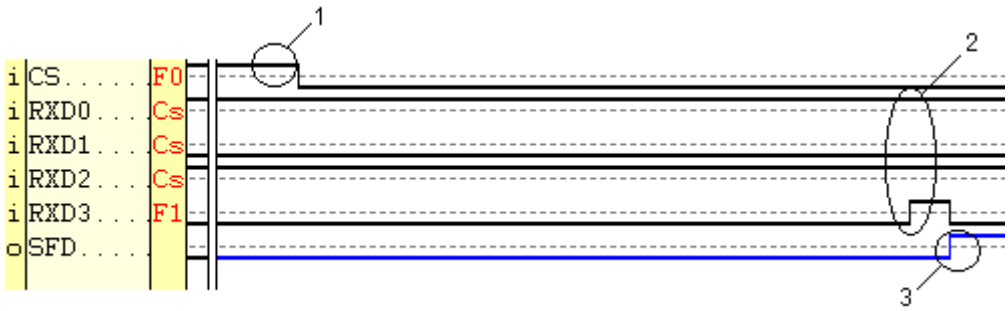
ภาคผนวก ข รายละเอียดของวงจรและผลการจำลองการทำงาน

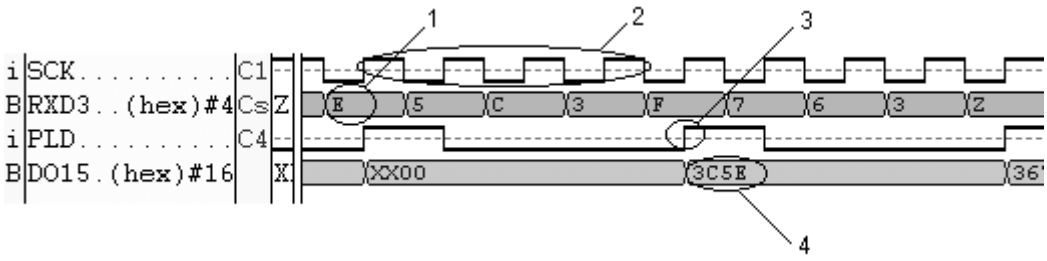
ข.1 ผลการจำลองการทำงานวงจร *SFD_Det* แสดงดังภาพประกอบ ข-1 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-1 ผลการจำลองการทำงานของวงจร *SFD_Det*

1. วงจร *SFD_Det* จะถูกรีเซ็ตเมื่อสัญญาณ *CS* มีสถานะเป็นลอจิก 1
2. เมื่อข้อมูล SFD ซึ่งมีลำดับการส่งของบิตดังนี้คือ 10101011 โดยบิต 1 ซ้ายมือสุดคือบิต LSB ถูกส่งมาในสัญญาณ *RXD[3:0]*
3. เมื่อวงจร *SFD_Det* ตรวจพบสัญญาณ SFD จะเปลี่ยนสถานะของสัญญาณเอาท์พุต *SFD* จากสถานะลอจิก 0 เป็นลอจิก 1

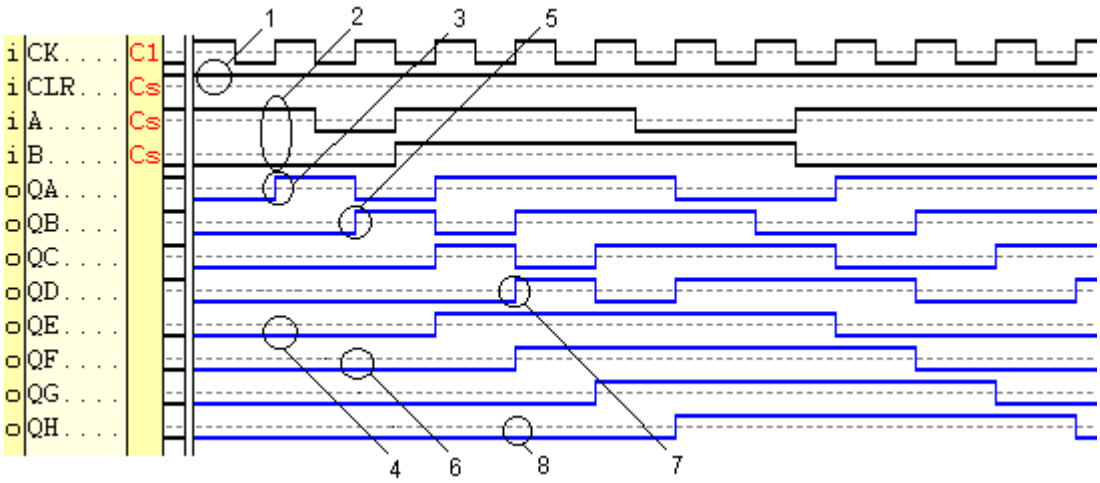
ข.2 ผลการจำลองการทำงานวงจร *S2P* แสดงดังภาพประกอบ ข-2 โดยสมมติให้ข้อมูลคือ 0x5E และ 0x3C ถูกส่งเข้ามาผ่านทางสัญญาณ *RXD[3:0]* ตามลำดับ ซึ่งสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-2 ผลการจำลองการทำงานของวงจร *S2P*

1. ข้อมูลจะถูกส่งทีละ 4 บิตโดยข้อมูล 4 บิตล่างจะถูกส่งออกมาก่อน ซึ่งในที่นี้ข้อมูล 4 บิตล่างของ 0x5E คือ 0xE
2. หลังจากสัญญาณนาฬิกาผ่านไป 4 ลูก สัญญาณอนุกรมจะถูกแปลงเป็นสัญญาณขนานขนาด 16 บิต
3. เมื่อสัญญาณ *PLD* ซึ่งก็คือสัญญาณ *RX_CEO* ที่ส่งมาจากวงจร *ADR_CNT* เปลี่ยนสถานะจากลอจิก 0 เป็นลอจิก 1
4. ข้อมูล 0x5E และ 0x3C ซึ่งถูกแปลงเป็นสัญญาณขนาน 16 บิตจะถูกค้างค่าไปออกที่เอาท์พุตของวงจร *S2P* และเนื่องจากลำดับการส่งข้อมูลอนุกรมผ่านทางสัญญาณ *RXD[3:0]* นั้นข้อมูลไบต์ต่ำจะถูกส่งเข้ามาก่อน ดังนั้นเอาท์พุตของวงจร *S2P* จึงมีค่าเท่ากับ 0x3C5E

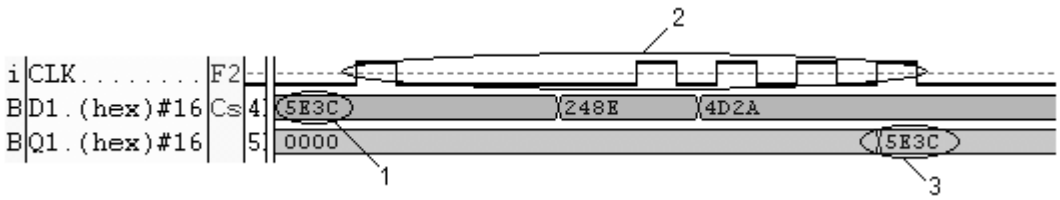
ข.3 ผลการจำลองการทำงานวงจร *MOD_164* ซึ่งเป็นวงจรที่ได้จากการดัดแปลงวงจร x74_164 เพื่อใช้สลับลำดับบิตข้อมูลขนาด 4 บิตที่ส่งผ่านทาง MII เป็นข้อมูลขนาด 16 บิต แสดงดังภาพประกอบ ข-3 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-3 ผลการจำลองการทำงานของวงจร *MOD_164*

1. เมื่อสัญญาณ *CLR* มีสถานะเป็นลอจิก 1 ข้อมูลที่อินพุตของวงจร MOD_164 จะสามารถเลื่อนบิตข้อมูลไปออกยังเอาท์พุตได้เมื่อพบสัญญาณนาฬิกาขาขึ้น
2. เมื่อพบสัญญาณนาฬิกาขาขึ้นลูกที่ 1 อินพุตที่ *A* และ *B* มีสถานะเป็นลอจิก 1 และ 0 ตามลำดับ
3. เมื่อพบสัญญาณนาฬิกาขาขึ้นลูกที่ 1 อินพุตที่ *A* จะไปปรากฏที่ *QA*
4. เมื่อพบสัญญาณนาฬิกาขาขึ้นลูกที่ 1 อินพุตที่ *B* จะไปปรากฏที่ *QE*
5. เมื่อพบสัญญาณนาฬิกาขาขึ้นลูกที่ 2 สัญญาณที่ *QA* จะเลื่อนบิตไปปรากฏที่ *QB*
6. เมื่อพบสัญญาณนาฬิกาขาขึ้นลูกที่ 2 สัญญาณที่ *QE* จะเลื่อนบิตไปปรากฏที่ *QF*
7. เมื่อพบสัญญาณนาฬิกาขาขึ้นลูกที่ 4 อินพุตที่ *A* จะไปปรากฏที่ *QD*
8. เมื่อพบสัญญาณนาฬิกาขาขึ้นลูกที่ 4 อินพุตที่ *B* จะไปปรากฏที่ *QH*

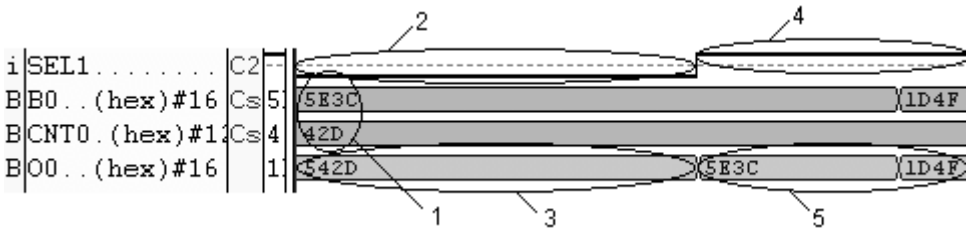
ข.4 ผลการจำลองการทำงานวงจร *INFOBUF* แสดงดังภาพประกอบ ข-4 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-4 ผลการจำลองการทำงานของวงจร *INFOBUF*

1. ข้อมูล 16 บิตมาถูกส่งมาที่อินพุตของวงจร *INFOBUF* ซึ่งก็คือขา *D[16:1]*
2. เมื่อมีสัญญาณพัลส์จากวงจร *ADR_CNT* ป้อนเข้ามา 5 ลูก
3. ข้อมูลชุดแรกซึ่งในที่นี้คือ 0x5E3C จะปรากฏที่เอาท์พุตของวงจร *INFOBUF* คือขา *Q[16:1]* หลังจากนั้นเมื่อมีสัญญาณพัลส์ป้อนเข้ามาจะทำให้ข้อมูลชุดที่ 2, 3, 4 และ 5 ที่จัดเก็บไว้ไปปรากฏที่เอาท์พุตของวงจร *INFOBUF* ตามลำดับ นั่นคือสัญญาณพัลส์ที่ป้อนเข้ามาเป็นทั้งสัญญาณที่ควบคุมการจัดเก็บข้อมูลและการอ่านค่าจากเอาท์พุตของวงจร *INFOBUF*

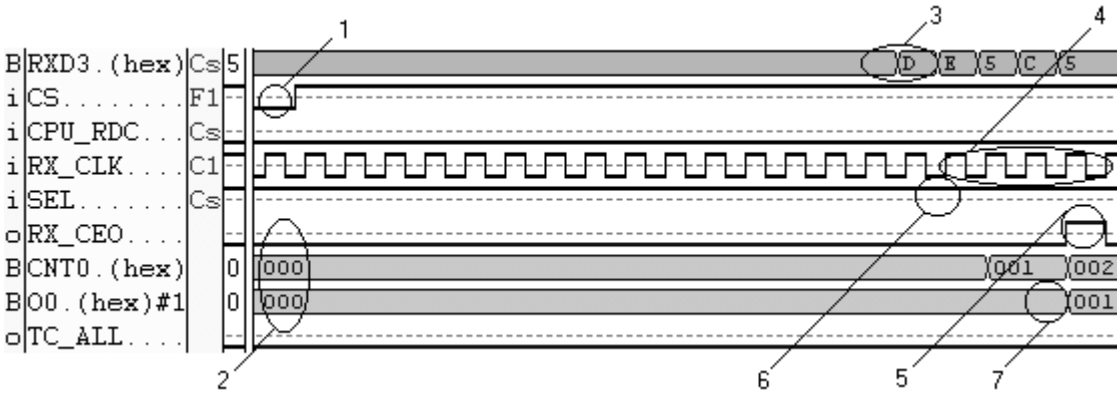
ข.5 ผลการจำลองการทำงานวงจร *INFOMUX* แสดงดังภาพประกอบ ข-5 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-5 ผลการจำลองการทำงานของวงจร *INFOMUX*

1. ข้อมูล 2 ชุดจากวงจร *INFOBUF* และวงจร *ADR_CNT* คือ *B[15:0]* และ *CNT[11:0]* ตามลำดับ ปรากฏที่อินพุตของวงจร *INFOMUX*
2. สัญญาณ *SEL1* มีสถานะเป็นลอจิก 0 เป็นการเลือกสัญญาณ *CNT[11:0]* ให้ออกที่เอาท์พุตของวงจร *INFOMUX*
3. ข้อมูลจากสัญญาณ *CNT[11:0]* คือ 0x42D ปรากฏที่เอาท์พุตของวงจร *INFOMUX* และเนื่องจากเอาท์พุตของวงจร *INFOMUX* มีขนาด 16 บิตซึ่งอีก 4 บิตบนเป็นการเซ็ตค่าคงที่คือ 0101 ดังนั้นเอาท์พุตที่ได้จึงมีค่าเท่ากับ 0x542D การเซ็ตค่าคงที่ 0101 นั้นมีไว้เพื่อใช้ตรวจสอบความถูกต้องของเอาท์พุตในภายหลัง
4. สัญญาณ *SEL1* มีสถานะเป็นลอจิก 1 เป็นการเลือกสัญญาณ *B[15:0]* ให้ออกที่เอาท์พุตของวงจร *INFOMUX*
5. ข้อมูลจากสัญญาณ *B[15:0]* คือ 0x5E3C และ 0x1D4F ปรากฏที่เอาท์พุตของวงจร *INFOMUX*

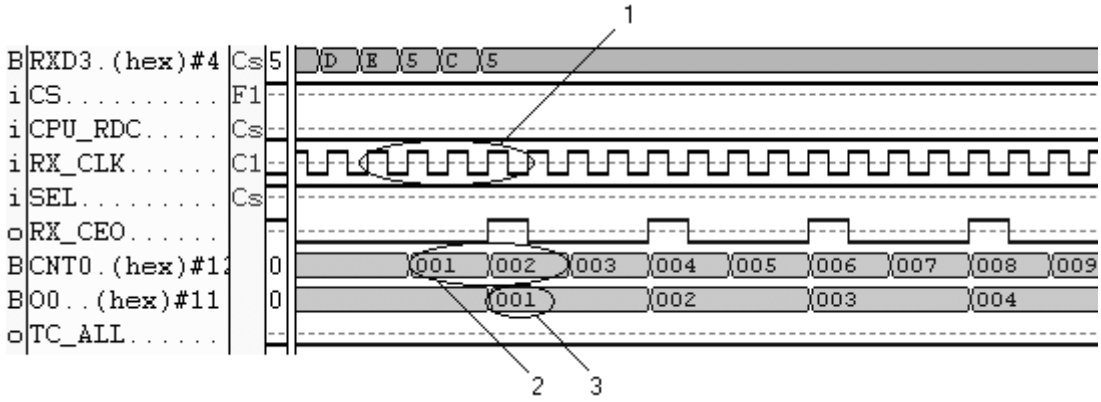
ข.6 ผลการจำลองการทำงานของวงจร *ADR_CNT* แบ่งออกเป็นผลการจำลองการทำงานในภาครับและภาคส่ง ผลการจำลองการทำงานในภาครับแสดงดังภาพประกอบ ข-6 ข-7 และ ข-8 ส่วนผลการจำลองการทำงานในภาคส่งแสดงดังภาพประกอบ ข-9 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-6 ผลการจำลองการทำงานของวงจร *ADR_CNT*

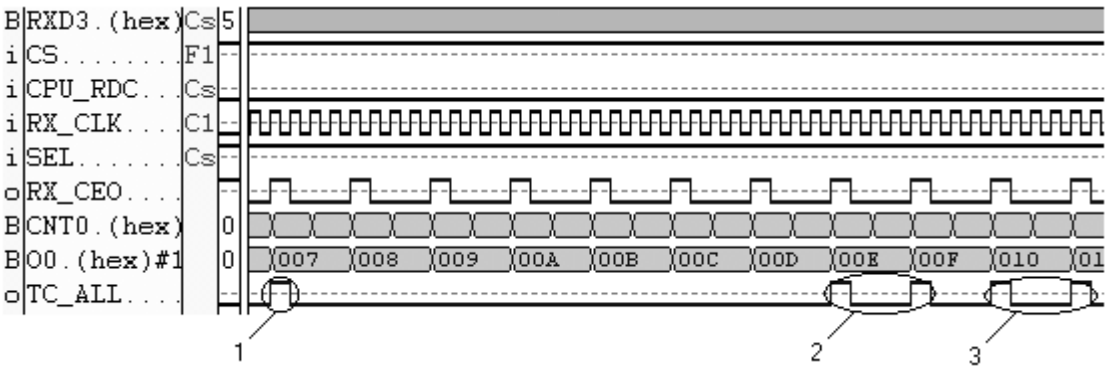
1. เมื่อสัญญาณ *CS* มีสถานะเป็นลอจิก 0 เป็นการรีเซ็ตวงจร *ADR_CNT*
2. หลังการรีเซ็ตวงจร สัญญาณเอาท์พุต *RX_CEO* มีสถานะเป็นลอจิก 0 และสัญญาณ *CNT[11:0]* ซึ่งเป็นสัญญาณบอกจำนวนไบต์ของแพ็คเกตที่ได้รับมีค่าเท่ากับ 0x000
3. เมื่อข้อมูล SFD ถูกส่งเข้ามาในสัญญาณ *RXD[3:0]*
4. เมื่อพบ SFD สัญญาณอนุกรม 4 บิตจะถูกแปลงเป็นสัญญาณขนาน 16 บิต หลังจากสัญญาณนาฬิกาผ่านไป 4 ลูก
5. เมื่อสัญญาณนาฬิกาผ่านไป 4 ลูก สัญญาณ *RX_CEO* จะเกิดพัลส์ขึ้น เพื่อใช้ในการโหลดข้อมูลแบบขนาน (Parallel Load) ซึ่งผ่านการแปลงจากอนุกรมเป็นขนานไปไว้ในบัสข้อมูลของหน่วยความจำหลัก
6. ขณะที่วงจรภาครับกำลังทำงาน สัญญาณ *SEL* ซึ่งก็คือสัญญาณ *RX_GNT* ที่วงจร *Arbiter* ส่งมา จะมีสถานะเป็นลอจิก 1 เพื่อเลือกแอดเดรสของวงจรภาครับให้ปรากฏที่ขา O[10:0] ซึ่งเป็นเอาท์พุตของวงจร *ADR_CNT*

7. สัญญาณเอาท์พุต *O[10:0]* ถูกควบคุมโดยสัญญาณ *SEL* ซึ่งเมื่อสัญญาณ *SEL* มีสถานะเป็นลอจิก 1 นั้นสัญญาณเอาท์พุต *O[10:0]* คือสัญญาณแอดเดรส 11 บิตล่างในการจัดเก็บข้อมูล 16 บิต เป็นสัญญาณเดียวกับ *CNT [11:1]* ซึ่งสัญญาณแอดเดรสแรกคือ 0x000



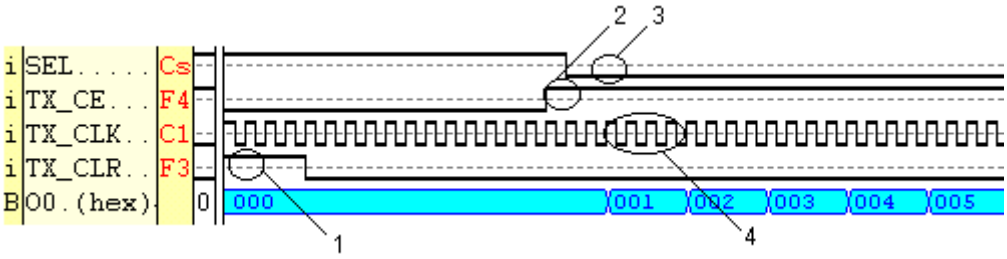
ภาพประกอบ ข-7 ผลการจำลองการทำงานของวงจร *ADR_CNT* (ต่อ)

1. เมื่อสัญญาณนาฬิกา *RX_CLK* ผ่านไป 4 ลูก
2. นับจำนวนไบต์ของแพ็คเกตได้เท่ากับ 2 ไบต์
3. แอดเดรสชี้ตำแหน่งหน่วยความจำจะเลื่อนไปยังตำแหน่งถัดไป



ภาพประกอบ ข-8 ผลการจำลองการทำงานของวงจร *ADR_CNT* (ต่อ)

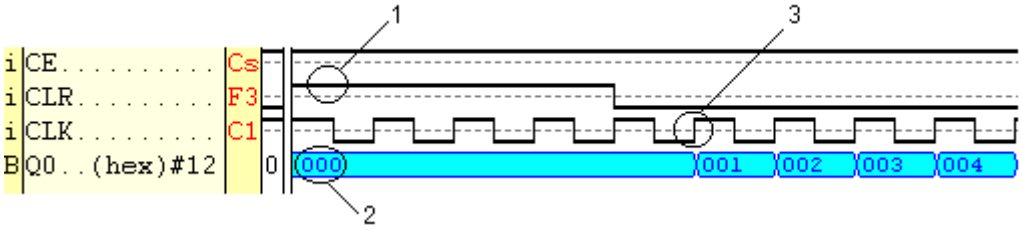
1. สัญญาณพัลส์ *RX_CEO* ลูกที่ 7 จะทำให้เกิดสัญญาณพัลส์ *TC_ALL* เพื่อค้างค่าไบต์ที่ 13 และ 14 ของแพ็คเกต ซึ่งเป็นไบต์บอกชนิดของแพ็คเกต
2. สัญญาณพัลส์ *RX_CEO* ลูกที่ 14 และ 15 จะทำให้เกิดสัญญาณพัลส์ *TC_ALL* เพื่อค้างค่าไบต์ที่ 27, 28, 29 และ 30 ของแพ็คเกต ซึ่งเป็นไอพีแอดเดรสต้นทางของแพ็คเกต
3. สัญญาณพัลส์ *RX_CEO* ลูกที่ 16 และ 17 จะทำให้เกิดสัญญาณพัลส์ *TC_ALL* เพื่อค้างค่าไบต์ที่ 31, 32, 33 และ 34 ของแพ็คเกต ซึ่งเป็นไอพีแอดเดรสปลายทางของแพ็คเกต



ภาพประกอบ ข-9 ผลการจำลองการทำงานของวงจร *ADR_CNT* (ต่อ)

1. สัญญาณ *TX_CLR* มีสถานะเป็นลอจิก 1 เป็นการรีเซ็ตค่าพอยน์เตอร์
2. สัญญาณ *TX_CE* มีสถานะเป็นลอจิก 1 หมายถึงวงจรภาคส่งกำลังทำงาน พอยน์เตอร์สามารถปรับค่าได้
3. สัญญาณ *SEL* มีสถานะเป็นลอจิก 0 เป็นการเลือกเอาท์พุตเป็นค่าแอดเดรสของวงจรภาคส่ง
4. เมื่อสัญญาณนาฬิกา *TX_CLK* ผ่านไป 4 ลูก พอยน์เตอร์จะปรับค่าเพิ่มขึ้น 1 ค่า

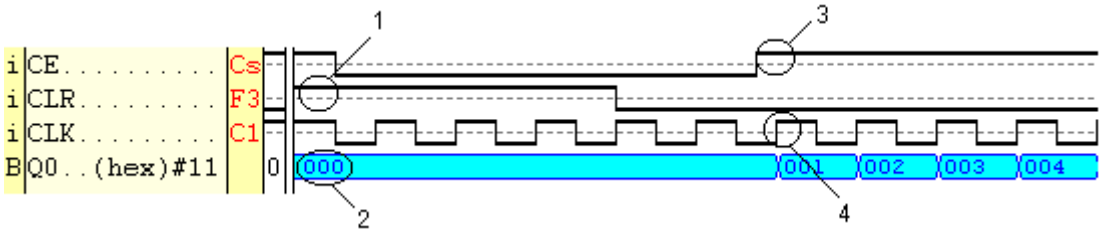
ข.7 ผลการจำลองการทำงานวงจร *CNT12* ซึ่งเป็นวงจรเคาน์เตอร์ขนาด 12 บิตใช้ในการนับเพื่อระบุตำแหน่งแอดเดรสที่จัดเก็บแพ็คเกตของวงจรภาครับ ผลการจำลองการทำงานแสดงดังภาพประกอบ ข-10 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-10 ผลการจำลองการทำงานของวงจร CNT12

1. สัญญาณ *CLR* มีสถานะลอจิกเป็น 1 เป็นการเคลียร์ค่าเคาน์เตอร์ โดยจะไม่สนใจสถานะของสัญญาณ *CE*
2. เอาท์พุตของเคาน์เตอร์หลังจากถูกเคลียร์ค่ามีค่าเท่ากับ 0x000
3. ขณะที่สัญญาณ *CE* มีสถานะเป็นลอจิก 1 ทุกครั้งที่มีสัญญาณนาฬิกาขาขึ้นป้อนเข้ามา เคาน์เตอร์จะทำการเพิ่มค่าไปทีละ 1

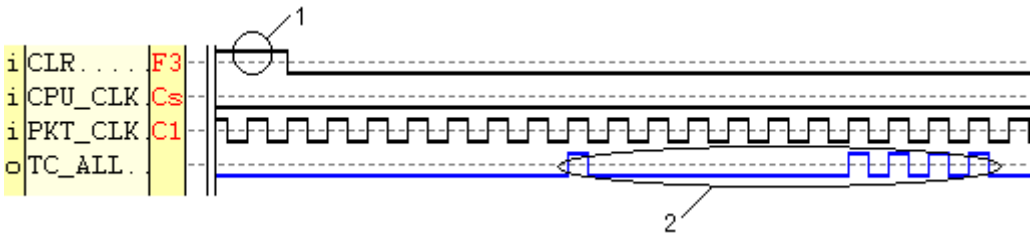
ข.8 ผลการจำลองการทำงานวงจร *CNT11* ซึ่งเป็นวงจรเคาน์เตอร์ขนาด 11 บิตใช้ในการนับเพื่อระบุตำแหน่งแอดเดรสของแพ็คเกตของวงจรภาคส่ง ผลการจำลองการทำงานแสดงดังภาพประกอบ ข-11 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-11 ผลการจำลองการทำงานของวงจร *CNT11*

1. สัญญาณ *CLR* มีสถานะลอจิกเป็น 1 เป็นการเคลียร์ค่าเคาน์เตอร์ โดยจะไม่สนใจสถานะของสัญญาณ *CE*
2. เอาท์พุตของเคาน์เตอร์หลังจากถูกเคลียร์ค่ามีค่าเท่ากับ 0x000
3. เมื่อสัญญาณ *CE* มีสถานะเป็นลอจิก 1 เคาน์เตอร์จะสามารถนับได้
4. ขณะที่สัญญาณ *CE* มีสถานะเป็นลอจิก 1 ทุกครั้งที่มีสัญญาณนาฬิกาขาขึ้นป้อนเข้ามา เคาน์เตอร์จะทำการเพิ่มค่าไปทีละ 1

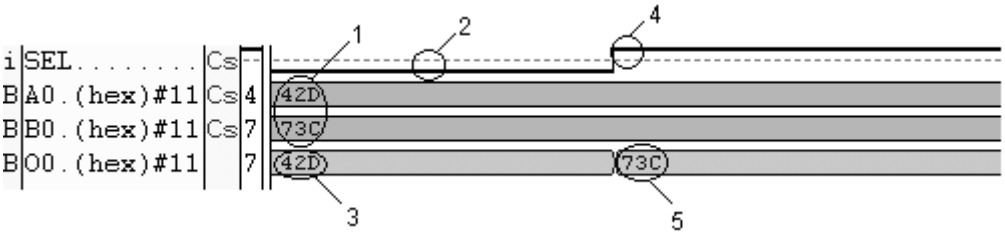
ข.9 ผลการจำลองการทำงานวงจร *INFOCTRL* ซึ่งเป็นวงจรที่ทำหน้าที่สร้างสัญญาณพัลส์เพื่อควบคุมการจัดเก็บข้อมูลของวงจร *INFOBUF* ผลการจำลองการทำงานแสดงดังภาพประกอบ ข-12 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-12 ผลการจำลองการทำงานของวงจร *INFOCTRL*

1. สัญญาณ *CLR* มีสถานะลอจิกเป็น 1 เป็นการรีเซ็ตวงจรฟลิปฟลอปภายใน
2. หลังจากสัญญาณ *CLR* เปลี่ยนสถานะเป็นลอจิก 0 วงจร *INFOCTRL* จะสร้างสัญญาณพัลส์เมื่อพบสัญญาณนาฬิกาขาขึ้นลูกที่ 7, 14, 15, 16 และ 17 ซึ่งจะตรงกับตำแหน่งไบต์ข้อมูลไบต์ที่ 13, 14 และ 27 ถึง 34 อันเป็นตำแหน่งของชนิดของแพ็คเกต ไอพีแอดเดรสต้นทาง และ ไอพีแอดเดรสปลายทางตามลำดับ

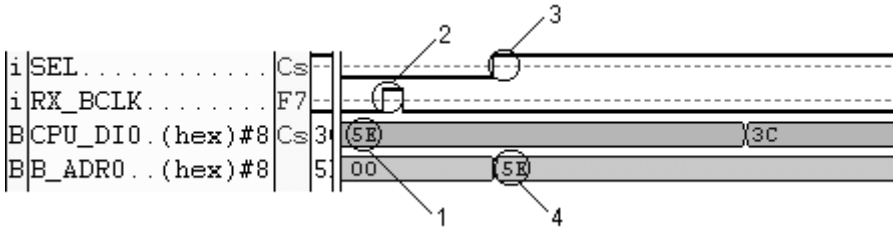
ข.10 ผลการจำลองการทำงานวงจร *MUX11* ซึ่งเป็นวงจรมัลติเพล็กเซอร์ขนาด 11 บิต แสดงดังภาพประกอบ ข-13 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-13 ผลการจำลองการทำงานของวงจร *MUX11*

1. ข้อมูล 2 ชุดคือ *A[10:0]* และ *B[10:0]* ตามลำดับ ปรากฏที่อินพุตของวงจร *MUX11*
2. สัญญาณ *SEL* มีสถานะเป็นลอจิก 0 เป็นการเลือกสัญญาณ *A[10:0]* ให้ออกที่เอาท์พุตของวงจร *MUX11*
3. ข้อมูลจากสัญญาณ *A[10:0]* คือ 0x42D ปรากฏที่เอาท์พุตของวงจร *MUX11*
4. สัญญาณ *SEL* มีสถานะเป็นลอจิก 1 เป็นการเลือกสัญญาณ *B[10:0]* ให้ออกที่เอาท์พุตของวงจร *MUX11*
5. ข้อมูลจากสัญญาณ *B[10:0]* คือ 0x73C ปรากฏที่เอาท์พุตของวงจร *MUX11*

ข.11 ผลการจำลองการทำงานวงจร *CPU_REGS* แบ่งออกเป็นผลการจำลองการทำงานในภาครับและภาคส่ง ผลการจำลองการทำงานในภาครับแสดงดังภาพประกอบ ข-14 และผลการจำลองการทำงานในภาคส่งแสดงดังภาพประกอบ ข-15 ข-16 และ ข-17 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



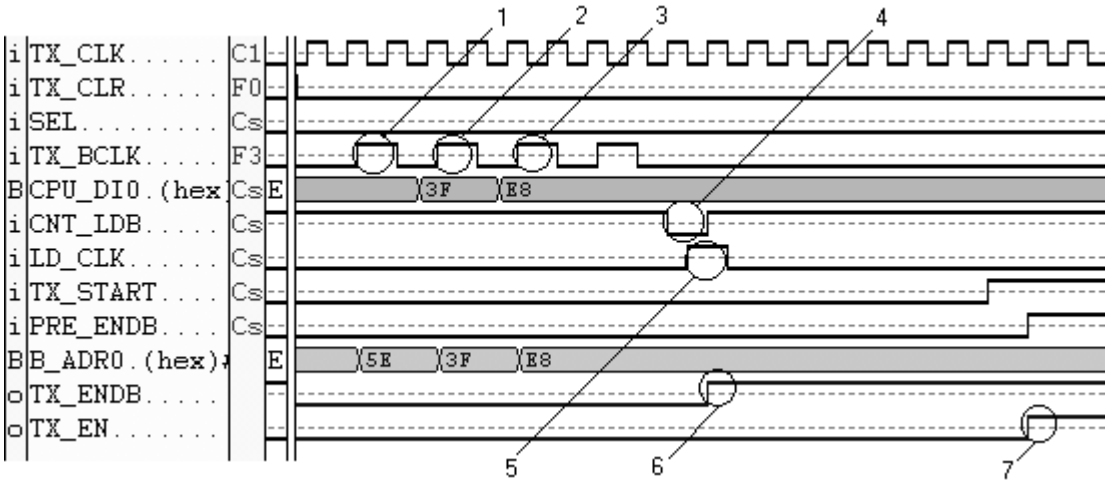
ภาพประกอบ ข-14 ผลการจำลองการทำงานของวงจร *CPU_REGS*

1. ไมโครโปรเซสเซอร์ส่งค่า *Base Address* ซึ่งในที่นี้คือค่า 0x5E มายังวงจร *CPU_REGS*
2. สัญญาณ *RX_BCLK* เปลี่ยนสถานะจากลอจิก 0 เป็นลอจิก 1 เป็นการค้างค่า *Base Address* ไว้ในรีจิสเตอร์ภายใน
3. สัญญาณ *SEL* มีสถานะเป็นลอจิก 1 เป็นการเลือกเอาท์พุตเป็นค่า *Base Address* ของวงจรภาครับ
4. ค่า *Base Address* ของวงจรภาครับที่อินพุตของวงจร *CPU_REGS* จะถูกส่งไปออกที่เอาท์พุตของวงจร *CPU_REGS*

ภาพประกอบ ข-15 ผลการจำลองการทำงานของวงจร *CPU_REGS* (ต่อ)

1. ไมโครโปรเซสเซอร์ส่งค่า *Base Address* ซึ่งในที่นี้คือ 0x5E มาให้วงจร *CPU_REGS*

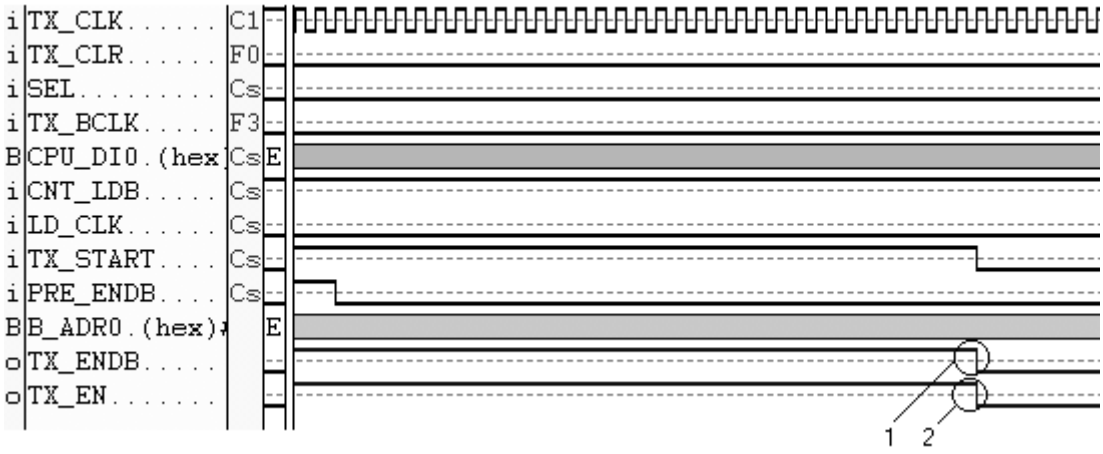
2. ไมโครโปรเซสเซอร์ส่งสัญญาณ *TX_BCLK* มาควบคุมให้วงจร *CPU_REGS* ค้างค่า *Base Address* ไว้ในรีจิสเตอร์ภายใน
3. สัญญาณ *SEL* มีสถานะเป็นลอจิก 0 เป็นการเลือกเอาท์พุตเป็นค่า *Base Address* ของวงจรภาคส่ง
4. ค่า *Base Address* ของวงจรภาคส่งที่อินพุตของวงจร *CPU_REGS* จะมาปรากฏที่เอาท์พุตของวงจร *CPU_REGS*



ภาพประกอบ ข-16 ผลการจำลองการทำงานของวงจร *CPU_REGS* (ต่อ)

1. สัญญาณพัลส์ *TX_BCLK* เป็นสัญญาณที่ไมโครโปรเซสเซอร์ส่งมาเพื่อใช้ควบคุมการค้างค่าเริ่มต้นในการนับไว้ในฟลิปฟลอปภายใน โดยพัลส์แรกจะทำการค้างค่า 1 ไบต์แรกไว้ที่อินพุตของวงจร *BYTECNT* ซึ่งก็คือ 0x3F
2. สัญญาณพัลส์ *TX_BCLK* ลูกที่ 2 ทำการโหลดค่า 4 บิตบนของค่าเริ่มต้นในการนับมาไว้ที่อินพุตของเคาน์เตอร์ขนาด 12 บิต และโหลดข้อมูลไบต์ที่ 2 ซึ่งก็คือ 0xE8 มาไว้ที่อินพุตของวงจร *BYTECNT*
3. สัญญาณพัลส์ *TX_BCLK* ลูกที่ 3 ทำการโหลดค่า 8 บิตล่างของค่าเริ่มต้นในการนับมาไว้ที่อินพุตของเคาน์เตอร์ขนาด 12 บิต
4. ไมโครโปรเซสเซอร์ส่งสัญญาณ *CNT_LDB* ที่มีสถานะเป็นลอจิก 0 เพื่อควบคุมการโหลดค่าเริ่มต้นในการนับไปไว้ในเคาน์เตอร์ขนาด 12 บิต

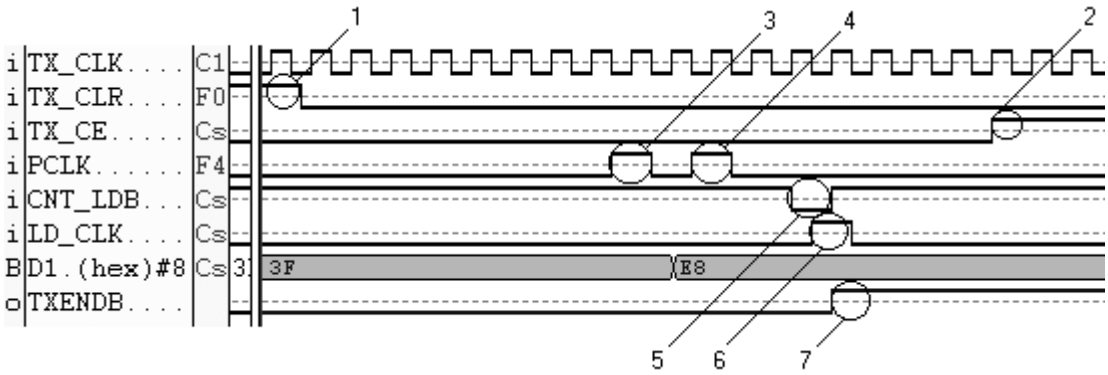
5. เมื่อไมโครโปรเซสเซอร์ส่งสัญญาณพัลส์ *LD_CLK* เคาน์เตอร์จะทำการโหลดค่าเริ่มต้นในการนับ ซึ่งในที่นี้ค่าเริ่มต้นในการนับคือ 0xFE8
6. สัญญาณ *TX_ENDB* เปลี่ยนสถานะเป็นลอจิก 1 เมื่อพบสัญญาณขาขึ้นแรกของสัญญาณนาฬิกา *TX_CLK* หลังจากทำการโหลดค่าเริ่มต้นในการนับ
7. สัญญาณ *TX_EN* เปลี่ยนสถานะเป็นลอจิก 1 เพื่อบอกให้ทราบว่ามีการส่งข้อมูลใน *TXD[3:0]*



ภาพประกอบ ข-17 ผลการจำลองการทำงานของวงจร *CPU_REGS* (ต่อ)

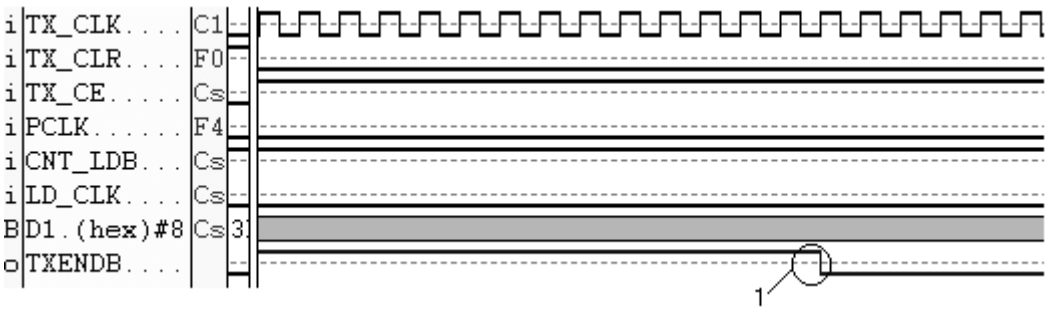
1. เมื่อเคาน์เตอร์นับจำนวนข้อมูลที่ส่งออกจนครบแล้วสัญญาณ *TX_ENDB* จะเปลี่ยนสถานะจากลอจิก 1 เป็น 0
2. เมื่อสิ้นสุดการส่งข้อมูลสัญญาณ *TX_EN* เปลี่ยนสถานะจากลอจิก 1 เป็น 0

ข.12 ผลการจำลองการทำงานวงจร *BYTECNT* แสดงดังภาพประกอบ ข-18 โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-18 ผลการจำลองการทำงานของวงจร *BYTECNT*

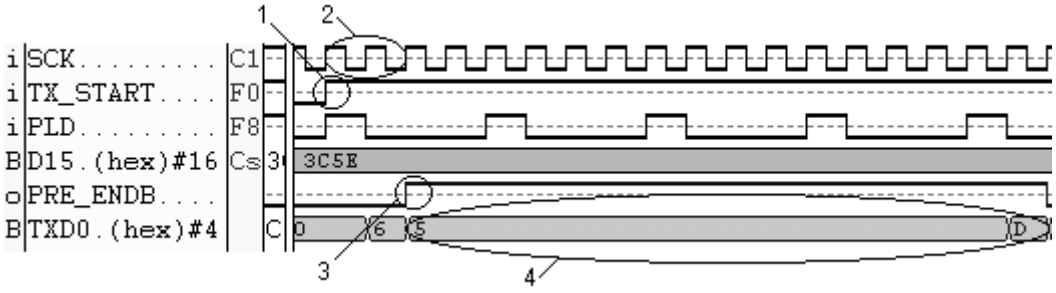
1. สัญญาณ *TX_CLR* มีสถานะเป็นลอจิก 1 เป็นการเคลียร์ค่าเคาน์เตอร์
2. สัญญาณ *TX_CE* มีสถานะเป็นลอจิก 1 เป็นการควบคุมให้เคาน์เตอร์สามารถเพิ่มค่าได้
3. สัญญาณ *PCLK* ใช้ในการโหลดค่าเริ่มต้นในการนับของแพ็คเกตมาไว้ที่อินพุตของเคาน์เตอร์ขนาด 12 บิต โดยสัญญาณ *PCLK* ลูกแรกจะโหลดค่า 4 บิตบน ซึ่งในที่นี้คือค่า 0xF
4. ส่วนสัญญาณ *PCLK* ลูกที่ 2 จะโหลดค่า 8 บิตล่าง ซึ่งในที่นี้คือค่า 0xE8
5. สัญญาณ *CNT_LDB* มีสถานะเป็นลอจิก 0 เป็นการควบคุมเคาน์เตอร์ให้สามารถโหลดค่าได้
6. เมื่อสัญญาณขาขึ้นของสัญญาณ *LD_CLK* เกิดขึ้น เคาน์เตอร์จะทำการโหลดค่าเริ่มต้นในการนับ ซึ่งในที่นี้ค่าเริ่มต้นของเคาน์เตอร์คือ 0xFE8
7. สัญญาณ *TX_ENDB* เปลี่ยนสถานะเป็นลอจิก 1 เมื่อพบสัญญาณนาฬิกาขาขึ้นแรกของสัญญาณนาฬิกา *TX_CLK* หลังจากทำการโหลดค่าเริ่มต้นในการนับ



ภาพประกอบ ข-19 ผลการจำลองการทำงานของวงจร *BYTECNT* (ต่อ)

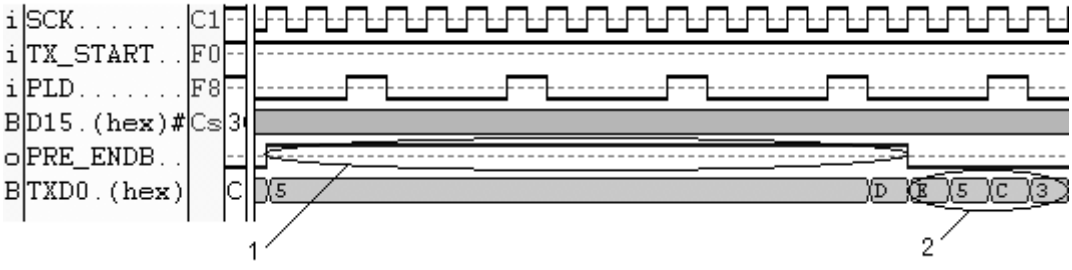
1. เคาน์เตอร์จะทำการนับเพิ่มทีละ 1 เมื่อพบสัญญาณขาขึ้นของสัญญาณนาฬิกาและเมื่อนับจนถึงค่า 0xFFF สัญญาณ *TXENDB* จะเปลี่ยนสถานะเป็นลอจิก 0

ข.13 ผลการจำลองการทำงานวงจร *P2S* แสดงดังภาพประกอบ ข-20 และ ข-21 ตามลำดับ โดยสามารถอธิบายลำดับเหตุการณ์ในผลการจำลองการทำงานได้ดังนี้



ภาพประกอบ ข-20 ผลการจำลองการทำงานของวงจร *P2S*

1. สัญญาณ *TX_START* ซึ่งส่งมาจากวงจร *Arbiter* เปลี่ยนสถานะจากลอจิก 0 เป็นลอจิก 1 เพื่อบอกให้เริ่มต้นการส่งข้อมูล
2. เมื่อสัญญาณนาฬิกาผ่านไป 2 ลูก
3. สัญญาณ *PRE_ENDB* จะเปลี่ยนสถานะจากลอจิก 0 เป็นลอจิก 1 เพื่อบอกช่วงเวลาการส่งสัญญาณ Preamble และ SFD
4. สัญญาณ Preamble และ SFD ถูกส่งออกทางสัญญาณข้อมูลของ MII



ภาพประกอบ ข-21 ผลการจำลองการทำงานของวงจร *P2S* (ต่อ)

1. สัญญาณ *PRE_ENDB* จะมีสถานะเป็นลอจิก 1 อยู่ 16 สัญญาณนาฬิกา
2. เมื่อส่งสัญญาณ Preamble SFD ถูกส่งออกจนครบแล้วข้อมูล 16 บิตซึ่งในที่นี้คือ 3C5E จะถูกแปลงและส่งออกผ่านทางสัญญาณข้อมูลของ MII

ภาคผนวก ค วงจร Arbiter วงจรรวม และอื่น ๆ

การกำหนดตำแหน่งขาของชิพเอฟพีจีเอ

ขา	ตำแหน่งขา	ประเภท	รายละเอียด
TX_CLK	4	อินพุต	เป็นขาสำหรับรับสัญญาณนาฬิกา TX_CLK ที่ได้รับผ่านทาง MII ซึ่งมีอัตราเร็วในการส่งข้อมูล 25 เมกกะบิตต่อวินาที
TXD[3:0]	5-8	เอาท์พุต	เป็นขาสำหรับใช้ส่งข้อมูลผ่านทาง MII โดยจะอ้างอิงสัญญาณนาฬิกา TX_CLK
TX_EN	10	เอาท์พุต	เป็นขาสำหรับส่งสัญญาณ TX_EN เพื่อบอกให้ทราบว่ามีการส่งข้อมูลผ่านทาง MII
ABR_CLR	11	อินพุต	เป็นขาที่ใช้สำหรับรีเซ็ตวงจร *Arbiter* โดยจะรีเซ็ตวงจรเมื่อสัญญาณที่ป้อนเข้าที่ขา *ABR_CLR* มีสถานะเป็นลอจิก 1
CS	12	อินพุต	เป็นขาที่ใช้สำหรับควบคุมการทำงานของวงจรภาครับ โดยจะรีเซ็ตวงจรภาครับเมื่อมีสถานะเป็นลอจิก 0 และจะทำงานเมื่อมีสถานะเป็นลอจิก 1 ซึ่งขานี้จะนำไปต่อกับสัญญาณ RX_DV ที่ได้รับผ่านทาง MII
CPU_RDC	13	อินพุต	เป็นขาที่ไมโครโปรเซสเซอร์ใช้ส่งสัญญาณพัลส์มาเพื่ออ่านค่าที่เก็บไว้ในวงจร *INFOBUF* ซึ่งค่าดังกล่าวจะถูกส่งออกมาที่เอาท์พุตของวงจร *INFOBUF* เมื่อสัญญาณ *CPU_RDC* เปลี่ยนสถานะจากลอจิก 0 เป็น 1

ขา	ตำแหน่งขา	ประเภท	รายละเอียด
IB[15:0]	15 - 24 และ 27 - 32	เอาท์พุต	เป็นขาสำหรับใช้ส่งสัญญาณข้อมูลที่ใช้ในการประมวลผลเพื่อจัดคิวแพ็คเกตซึ่งส่งมาจากวงจร *INFOBUF* หรือจำนวนไบต์ของแพ็คเกตซึ่งส่งมาจากวงจร *ADR_CNT* ขึ้นอยู่กับสถานะของสัญญาณ *SEL1* ที่ใช้ควบคุมการเลือกเอาท์พุตของมัลติเพล็กเซอร์ ถ้าสัญญาณ *SEL1* มีสถานะเป็นลอจิก 0 เป็นการเลือกเอาท์พุตจากวงจร *ADR_CNT* และถ้าสัญญาณ *SEL1* มีสถานะเป็นลอจิก 1 เป็นการเลือกเอาท์พุตจากวงจร *INFOBUF*
SEL1	33	อินพุต	เป็นขาที่ไมโครโปรเซสเซอร์ใช้สำหรับส่งสัญญาณควบคุมมัลติเพล็กเซอร์ของวงจร *INFOMUX* ให้เลือกเอาท์พุตสำหรับวงจรภาครับ ถ้าสัญญาณ *SEL1* มีสถานะเป็นลอจิก 0 จะเลือกเอาท์พุตจากวงจร *ADR_CNT* คือค่าจำนวนไบต์ของแพ็คเกตที่ได้รับ ถ้าสัญญาณ *SEL1* มีสถานะเป็นลอจิก 1 จะเลือกข้อมูลสำหรับประมวลผลเพื่อจัดคิวให้ปรากฏที่ขา *IB[15:0]*
LD_CLK	35	อินพุต	เป็นขาที่ไมโครโปรเซสเซอร์ใช้สำหรับส่งสัญญาณพัลส์มาค้างค่าเริ่มต้นในการนับของวงจรภาคส่ง
CNT_LDB	36	อินพุต	เป็นขาที่ไมโครโปรเซสเซอร์ใช้สำหรับส่งสัญญาณมาควบคุมการโหลดค่าเริ่มต้นในการนับของวงจร *CPU_REGS* ซึ่งวงจร *CPU_REGS* จะสามารถโหลดค่าได้เมื่อสัญญาณ *CNT_LDB* มีสถานะเป็นลอจิก 0

ขา	ตำแหน่งขา	ประเภท	รายละเอียด
CPU_DI [7:0]	39 – 46	อินพุต	เป็นขาที่ไมโครโปรเซสเซอร์ใช้เพื่อส่งค่า *Base Address* และค่าเริ่มต้นในการนับของวงจรภาคส่ง
TX_BLD	47	อินพุต	เป็นขาสำหรับไมโครโปรเซสเซอร์ส่งสัญญาณพัลส์มาเพื่อควบคุมการค้างค่า *Base Address* และค่าเริ่มต้นในการนับของวงจรภาคส่งโดยจะทำการค้างค่าเมื่อสัญญาณ *TX_BLD* เปลี่ยนสถานะจากลอจิก 0 เป็น 1
RX_BLD	57	อินพุต	เป็นขาสำหรับไมโครโปรเซสเซอร์ส่งสัญญาณพัลส์มาเพื่อควบคุมการค้างค่า *Base Address* ของวงจรภาครับโดยจะทำการค้างค่าเมื่อสัญญาณ *RX_BLD* เปลี่ยนสถานะจากลอจิก 0 เป็น 1
RAM_RW	59	เอาท์พุต	เป็นขาสำหรับส่งสัญญาณควบคุมการอ่านและเขียนหน่วยความจำหลัก ถ้าสัญญาณ *RAM_RW* มีสถานะเป็นลอจิก 0 หมายถึงการเขียนข้อมูล และลอจิก 1 หมายถึงการอ่านข้อมูลจากหน่วยความจำหลัก
RAM_CS	60	เอาท์พุต	เป็นขาสำหรับส่งสัญญาณเพื่อทำให้หน่วยความจำหลัก แอคทีพเมื่อสัญญาณ *RAM_CS* เป็นลอจิก 0
RAM_OE	61	เอาท์พุต	เป็นขาที่ใช้สำหรับควบคุมขาเอาท์พุตเอาท์พุตอีเนเบิลของหน่วยความจำหลัก

ขา	ตำแหน่งขา	ประเภท	รายละเอียด
BADR[7:0]	63 – 66 และ 68 – 71	เอาท์พุต	เป็นขาสำหรับให้ไมโครโปรเซสเซอร์ส่งสัญญาณ Base Address เพื่อระบุตำแหน่งแอดเดรส 8 บิตบนของแพ็คเกต ซึ่งขานี้จะถูกใช้งานร่วมกันระหว่างวงจรภาครับและวงจรภาคส่ง โดยขานี้จะถูกนำไปต่อกับบัสแอดเดรสของหน่วยความจำหลัก
ADR[10:0]	72 – 76 และ 80 - 85	เอาท์พุต	เป็นขาสำหรับส่งสัญญาณแอดเดรส 11 บิตล่างของแพ็คเกตไปยังหน่วยความจำหลัก
IO[15:0]	86 – 89 , 91 – 99 , 109 และ 111-112	อินพุต/เอาท์พุต	เป็นขาสำหรับเป็นอินพุตเอาท์พุตพอร์ตเพื่อใช้รับส่งข้อมูล ซึ่งขานี้จะต่ออยู่กับบัสข้อมูลของหน่วยความจำหลัก
RXD[3:0]	200 - 203	อินพุต	เป็นขาสำหรับใช้รับข้อมูลผ่านทาง MII โดยจะอ้างอิงสัญญาณนาฬิกา RX_CLK
RX_CLK	204	อินพุต	เป็นขาสำหรับรับสัญญาณนาฬิกา RX_CLK ที่ได้รับผ่านทาง MII ซึ่งมีอัตราเร็วในการรับข้อมูล 25 เมกกะบิตต่อวินาที