บทที่ 6
การทดลองการออกแบบวงจรตรรกะเชิงลำดับโดยใช้แนวคิดเชิงวัตถุ

เนื้อหาในบทนี้เกี่ยวข้องกับรายละเอียดการทดลองการออกแบบวงจรตรรกะเชิงลำดับ โดยใช้แนวคิดเชิงวัตถุซึ่งได้แก่ สภาพแวดล้อมการทดลอง วงจรที่ใช้ในการทดลอง ขั้นตอนการทดลอง และผลการทดลอง ซึ่งมีรายละเอียดดังต่อไปนี้

สภาพแวดล้อมการทดลอง

การทดลองการออกแบบวงจรตรรกะเชิงลำดับโดยใช้แนวคิดเชิงวัตถุ กระทำในเครื่องคอมพิวเตอร์ส่วนบุคคลที่มีหน่วยประมวลผลรุ่น Intel™ Pentium IV 3.0 กิกะเฮิร์ตซ์ (GHz) หน่วยความจำขนาด 1 GB ใช้ระบบปฏิบัติการรุ่น Microsoft Windows XP Professional และติดตั้งโปรแกรมเดลไฟล์รุ่น 7.0

วงจรที่ใช้ในการทดลอง

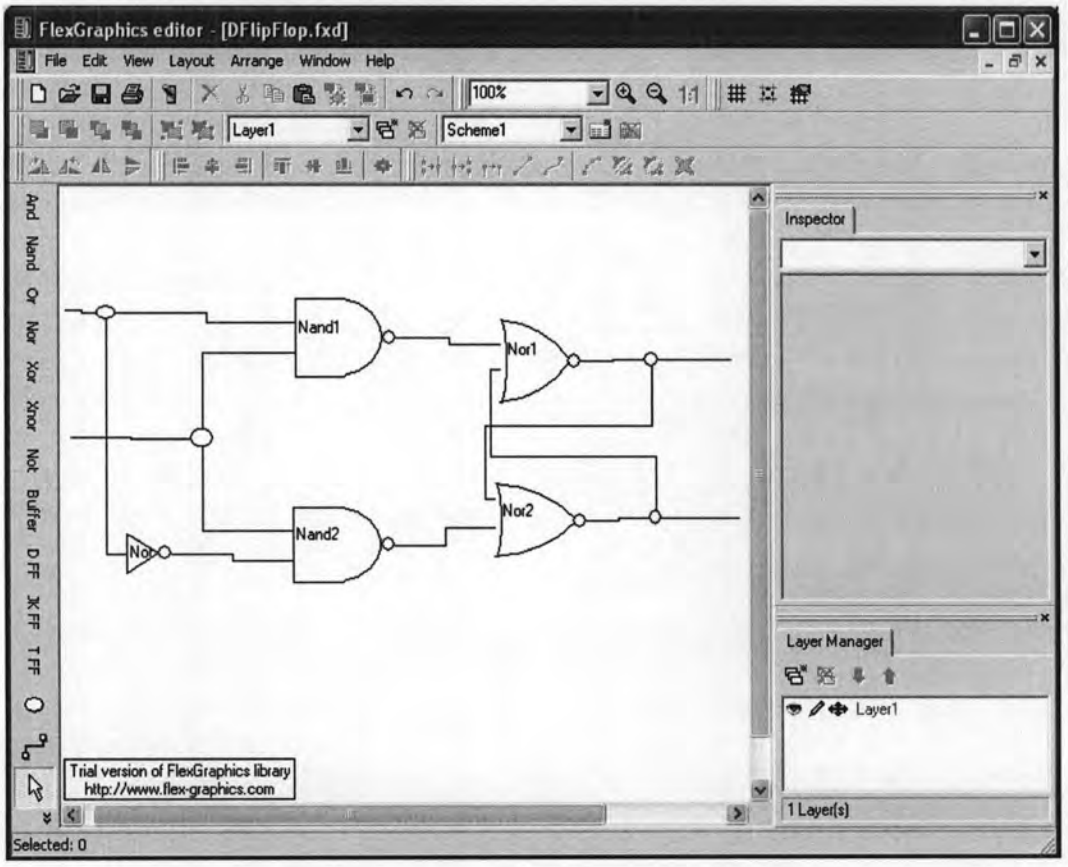
วงจรที่นำมาใช้ในการทดลองนั้น ได้เลือกตัวอย่างการสร้างวงจรในรูปแบบต่างๆ เพื่อทดสอบคุณสมบัติเชิงวัตถุทั้ง 3 คุณสมบัติ ดังนี้ คุณสมบัติการห่อหุ้ม ได้เลือกใช้วงจรตรรกะเชิงลำดับที่สร้างจากเกทต่างๆ และสามารถแทนการใช้ฟลิปฟลอปได้ และใช้วงจรบล็อกที่มีจำนวนวงจรบล็อกภายในมากกว่า 1 วงจร พร้อมการกำหนดการเชื่อมต่อสายสัญญาณกันระหว่างวงจรเหล่านั้น ซึ่งมีประโยชน์ในการประหยัดเวลาในการต้องสร้างวงจรซ้ำๆกันได้ คุณสมบัติการนำกลับมาใช้ ได้เลือกใช้วงจรบล็อกหลายๆ รูปแบบ ส่วนคุณสมบัติการถ่ายทอดจะได้แสดงไประหว่างการสร้างวงจรขั้นต้นแล้ว ทั้งนี้ยังได้ยกตัวอย่างการสร้างวงจรจากตัวอย่างวงจรมาตรฐาน ISCAS89 1 วงจร

ขั้นตอนการทดลองและผลการทดลอง

สำหรับขั้นตอนการทดลองและผลการทดลองของวงจรประเภทต่างๆ นั้น จะเรียงลำดับขั้นตอนการทดลองและผลการทดลองตามประเภทของวงจรดังนี้

1. วงจรตรรกะเชิงลำดับที่สร้างจากเกทต่างๆ และสามารถแทนการใช้ฟลิปฟลอปได้ จะนำตัวอย่างวงจรดีฟลิปฟลอปโดยสร้างขึ้นจากแนนด์เกทและนอร์เกทดังนี้

    1.1. ออกแบบวงจรดีฟลิปฟลอปโดยสร้างขึ้นจากแนนด์เกทและนอร์เกทได้ดังแสดงในรูปที่ 6.1



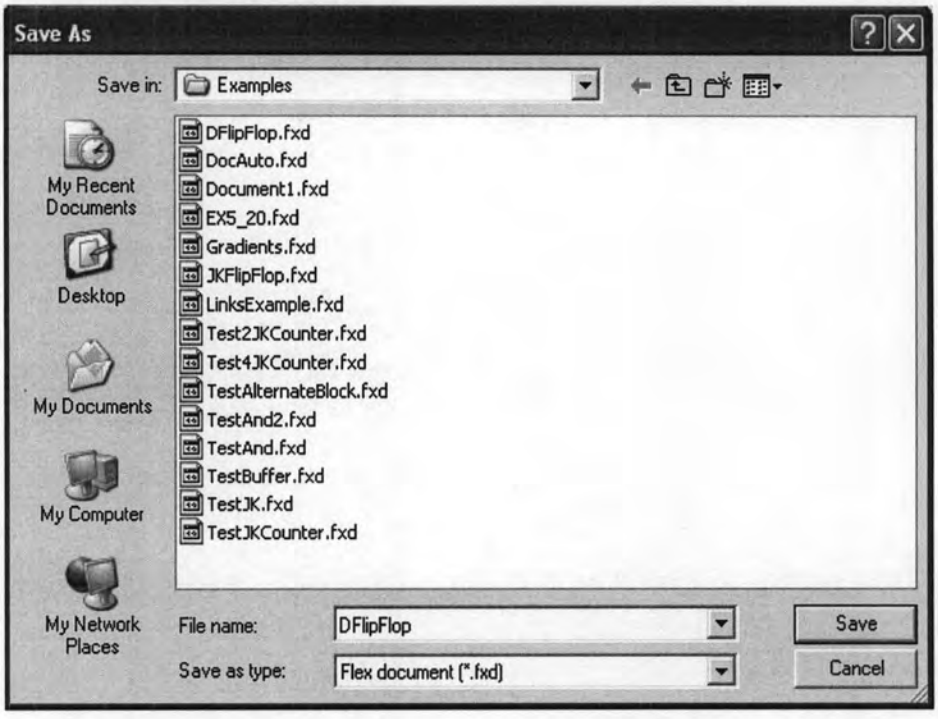
รูปที่ 6.1 แสดงการสร้างวงจรดีฟลิปฟลอปแบบกราฟิก

1.2. เซฟวงจรโดยเลือกเมนู File แล้วกด Save As ใส่ชื่อวงจร ในที่นี้ให้ชื่อวงจรว่า DFlipFlop เครื่องมือจะให้เลือกวงจรที่ได้รับการถ่ายทอดมา โดยมีตารางแสดงการถ่ายทอดให้เลือก ดังแสดงในรูปที่ 6.2 และ 6.3 หลังจากนั้นเครื่องมือจะทำการสังเคราะห์วงจรให้อัตโนมัติ และเก็บวงจรไว้ในรูปแบบของไฟล์เน็ตลิสต์ซึ่งเป็นไลบรารีของวงจร ไฟล์เน็ตลิสต์ที่ได้จากการสังเคราะห์วงจรจะแสดงในรูปที่ 6.4

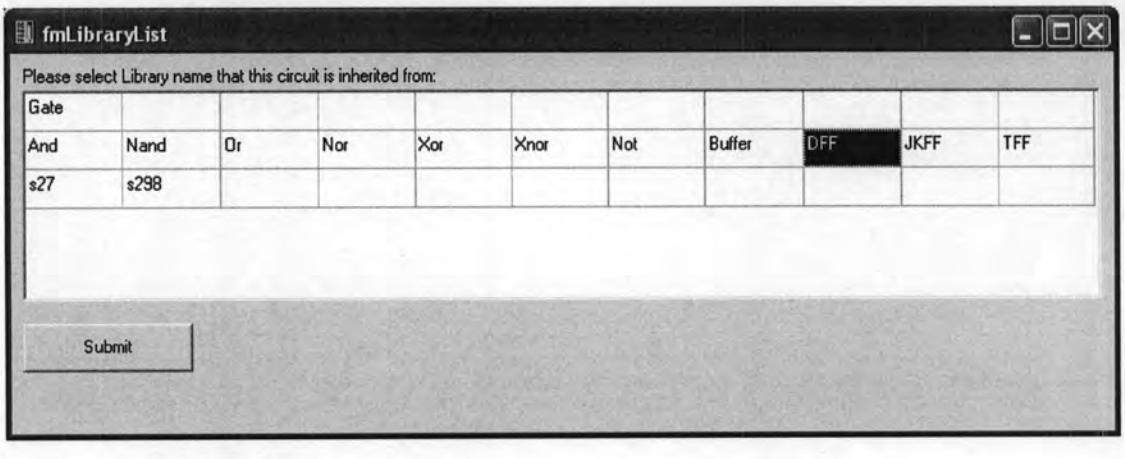
1.3. ทดสอบการสร้างวงจรจากวงจรดีฟลิปฟลอป โดยการสร้างวงจรบล็อกขึ้นมาโดยการกดปุ่ม เลือกวงจร DFlipFlop.netlist แล้ววาดลงบนพื้นที่วาดรูปวงจรดังรูป 6.5

1.4. เชื่อมต่อสายสัญญาณเข้าสู่อินพุตและเอาต์พุตของวงจรบล็อกโดยการเลือก และลากสายสัญญาณเชื่อมและกำหนดการเชื่อมต่อดังรูปที่ 6.6 - 6.8

1.5 เซฟวงจรและเลือกคลาสวงจรที่ถ่ายทอดมาเพื่อสังเคราะห์วงจร โดยให้ผลการสังเคราะห์วงจรเป็นไฟล์เน็ตลิสต์ดังรูปที่ 6.9



รูปที่ 6.2 แสดงการเก็บวงจรอยู่ในไลบรารี



รูปที่ 6.3 แสดงการเลือกคลาสที่ได้รับการถ่ายทอดจากตารางการถ่ายทอด

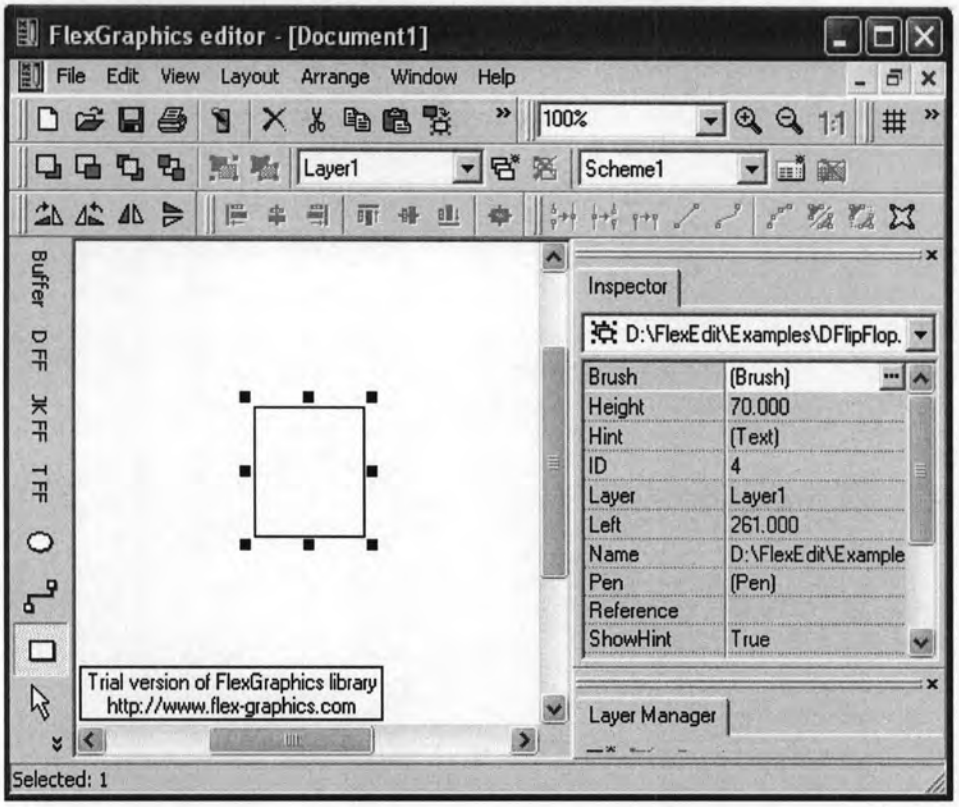
1.6 ทดลองสร้างวงจรแบบเดียวกันโดยใช้ฟลิปฟลอปดี ในการสร้าง ดังรูปที่ 6.10 – 6.12 และผลการสังเคราะห์วงจรจากการสร้างด้วยฟลิปฟลอปดีจะแสดงในรูปที่ 6.13

1.7 ผลการสังเคราะห์วงจรทั้งจากวงจรบล็อกตัวอย่างดีฟลิปฟลอปและจากวงจรที่สร้างจากดีฟลิปฟลอปโดยตรง พบว่าให้ส่วนที่ถูกต้องตรงกัน สังเกตได้จากบริเวณที่มีกรอบสีแดงล้อมรอบ ส่วนอื่นๆ จะเป็นรายละเอียดจากโครงสร้างการสร้างที่ไม่เหมือนกัน เพราะวงจรบล็อกตัวอย่างสร้างจากกลุ่มของเกทหลาย

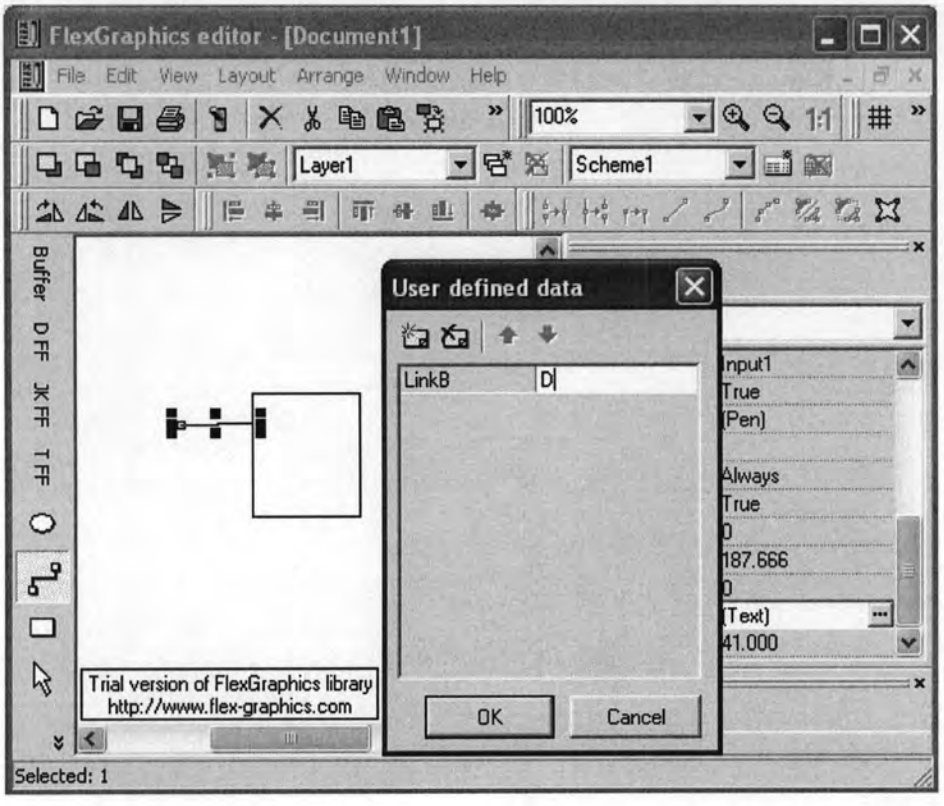
อุปกรณ์ แต่วงจรที่สร้างจากฟลิปฟลอปโดยตรงนั้นสร้างจากเพียงดีฟลิปฟลอป อุปกรณ์เดียว

```
1   # 2 inputs
2   # 2 outputs
3   # 1 inverters
4   # 4 gates (2 NANDs + 2 NORs)
5
6   INPUT(D)
7   INPUT(Clock)
8
9   OUTPUT(Q)
10  OUTPUT(Q2)
11
12  W1 = NOT(D)
13
14  W2 = NAND(D, Clock)
15  W3 = NAND(Clock, W1)
16
17  Q = NOR(W2, Q2)
18  Q2 = NOR(W3, Q)
19
```

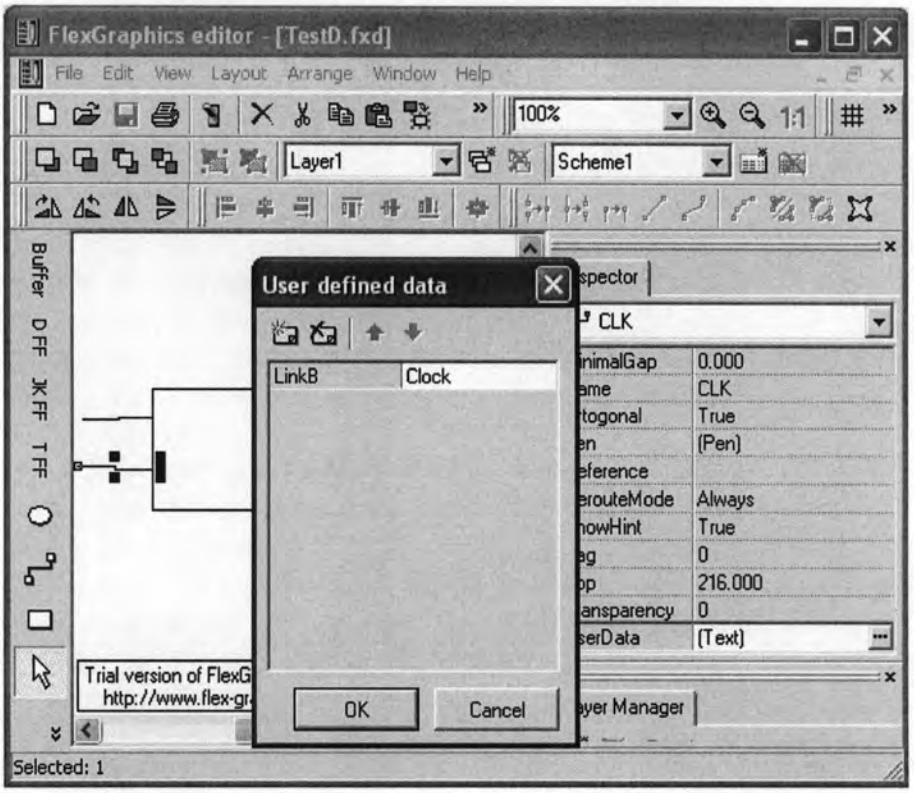
รูปที่ 6.4 แสดงผลจากการสังเคราะห์วงจรอยู่ในรูปแบบของไฟล์เน็ตลิสต์



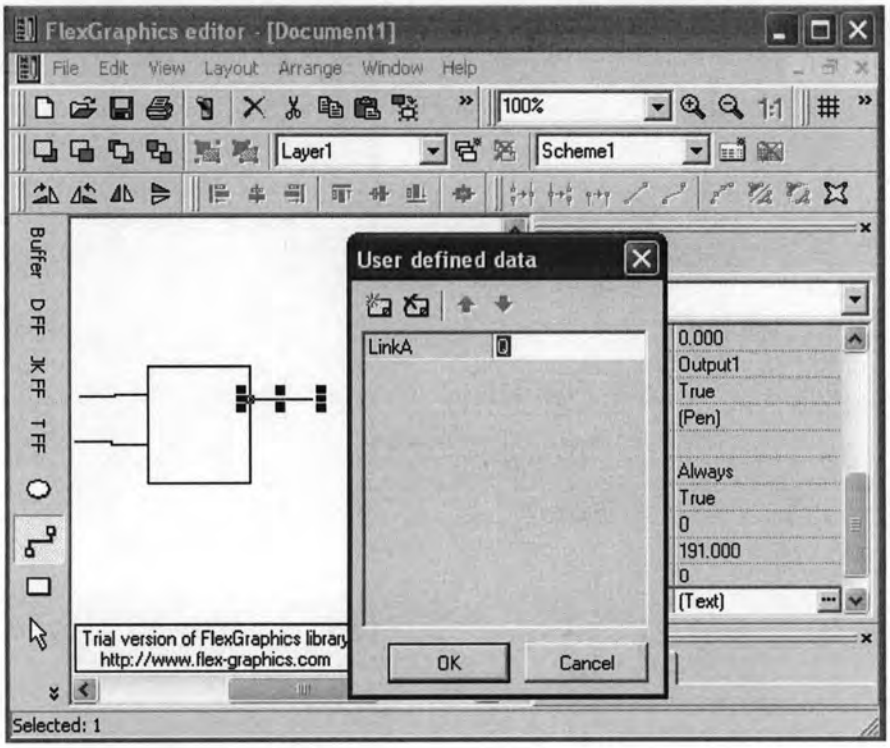
รูปที่ 6.5 แสดงการวาดวงจรบล็อก DFlipFlop ลงในเครื่องมือ



รูปที่ 6.6 แสดงการกำหนดการเชื่อมต่อระหว่างสายสัญญาณด้านอินพุตกับวงจรบล็อก



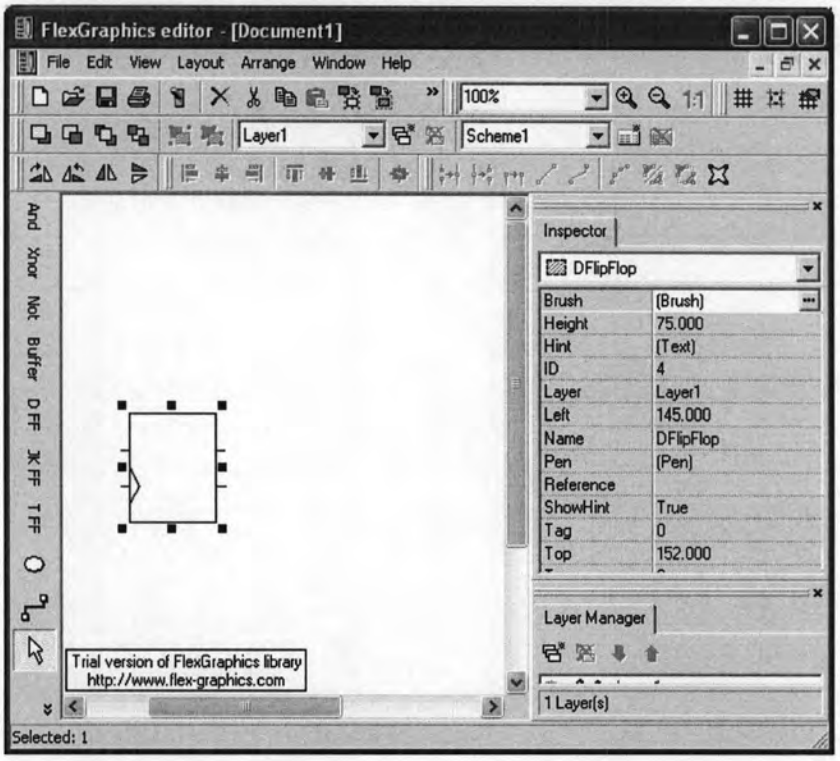
รูปที่ 6.7 แสดงการกำหนดการเชื่อมต่อระหว่างสายสัญญาณด้านอินพุตกับวงจรบล็อก



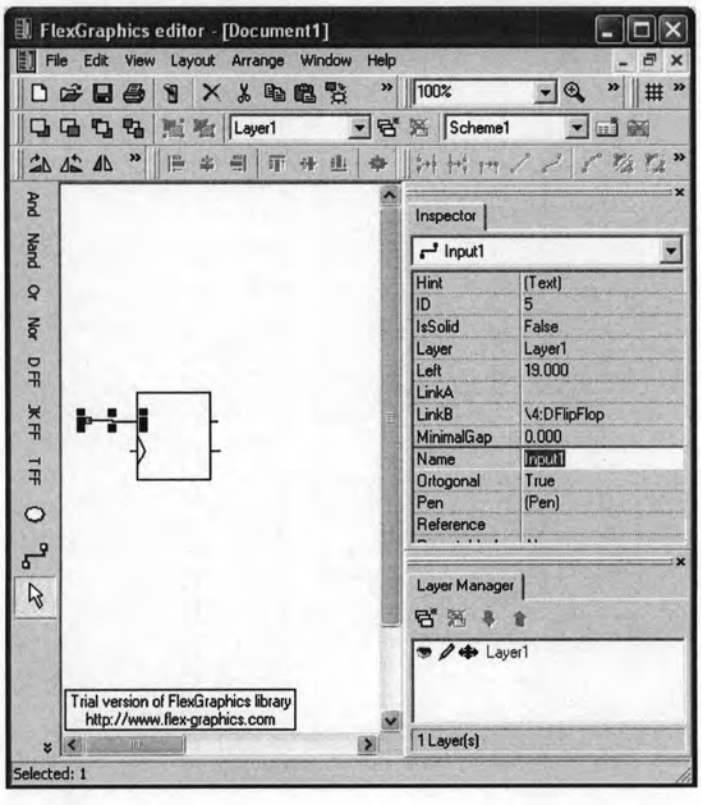
รูปที่ 6.8 แสดงการกำหนดการเชื่อมต่อระหว่างสายสัญญาณด้านเอาต์พุตกับวงจรบล็อก

```
1  # 2 inputs
2  # 2 outputs
3  # 1 inverters
4  # 4 gates (2 NANDs + 2 NORs)
5
6  INPUT(Input1)
7  INPUT(CLK)
8
9  OUTPUT(Output1)
10 OUTPUT(BLOCK4_Q2)
11
12 BLOCK4_W1 = NOT(Input1)
13
14 BLOCK4_W2 = NAND(Input1, CLK)
15 BLOCK4_W3 = NAND(CLK, BLOCK4_W1)
16
17 Output1 = NOR(BLOCK4_W2, BLOCK4_Q2)
18 BLOCK4_Q2 = NOR(BLOCK4_W3, Output1)
19
```

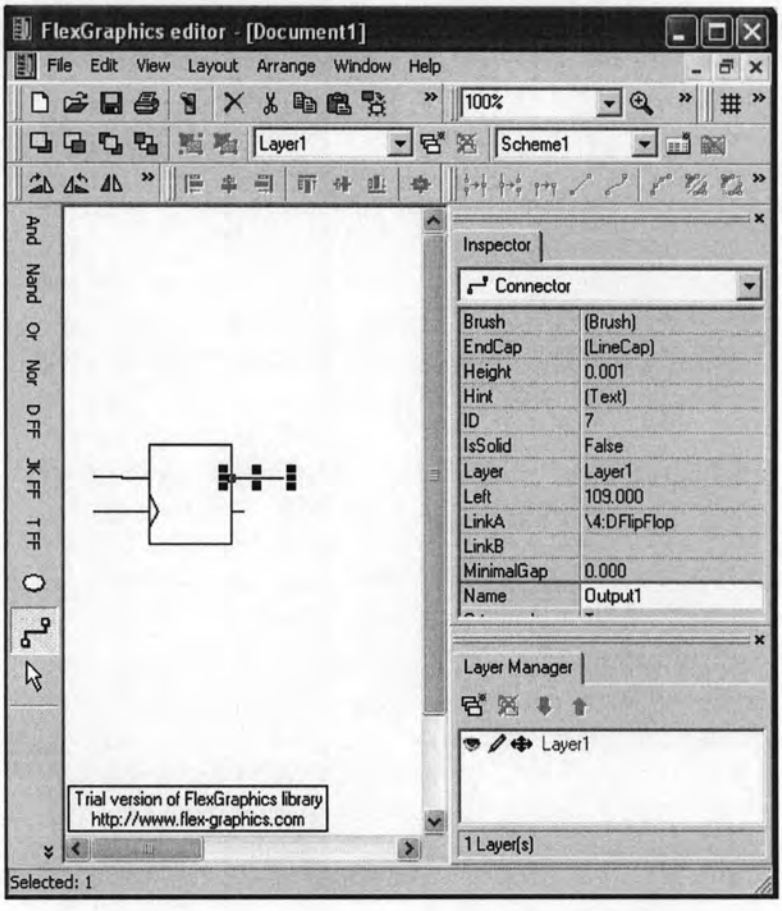
รูปที่ 6.9 แสดงผลการสังเคราะห์วงจร TestD.netlist



รูปที่ 6.10 แสดงการสร้างวงจรดีฟลิปฟลอปโดยใช้ฟลิปฟลอปดีโดยตรง



รูปที่ 6.11 แสดงการสร้างวงจรดีฟลิปฟลอปโดยใช้ฟลิปฟลอปดีโดยตรง(ต่อ)



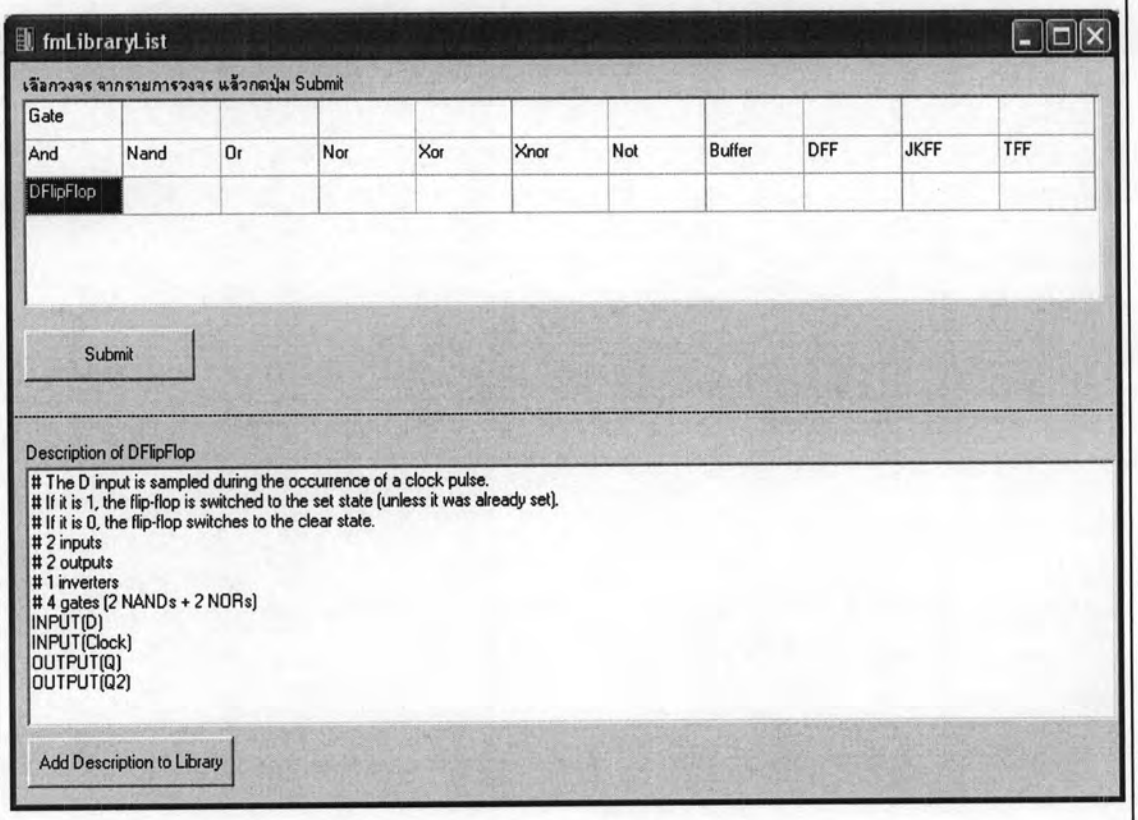
รูปที่ 6.12 แสดงการสร้างวงจรดีฟลิปฟลอปโดยใช้ฟลิปฟลอปดีโดยตรง(ต่อ)

```
1  # 2 inputs
2  # 1 outputs
3  # 1 D-type flipflops
4
5  INPUT(Input1)
6  INPUT(CLK)
7
8  OUTPUT(Output1)
9
10 Output1 = DFF(Input1, CLK)
11
```

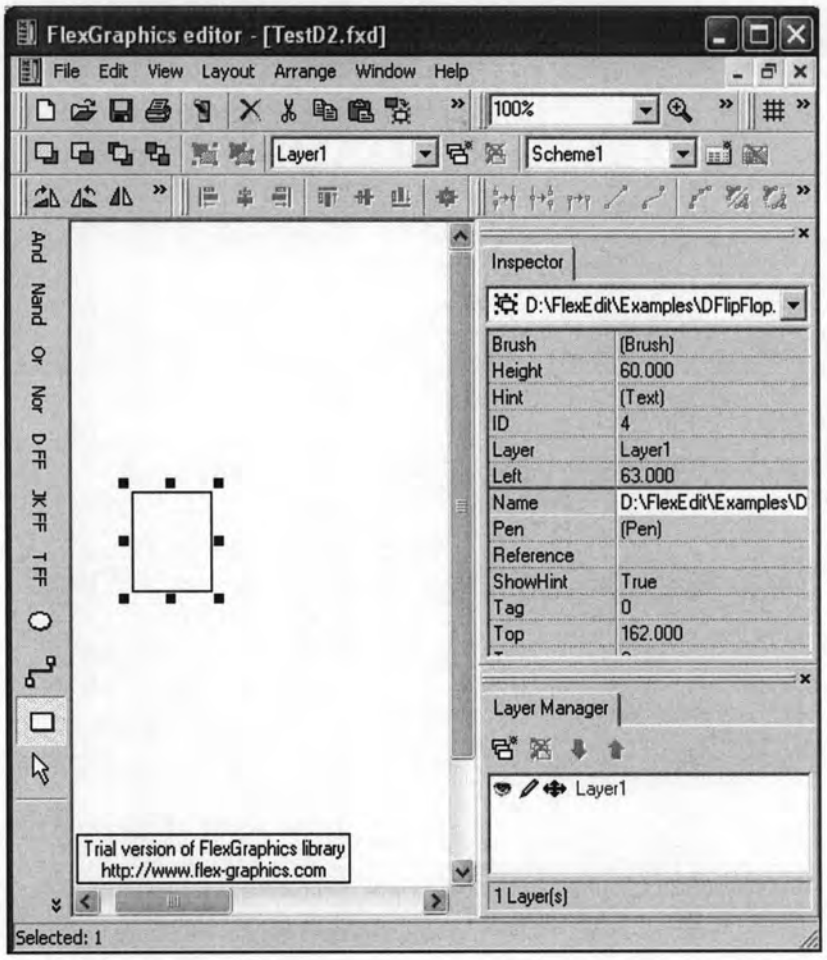
รูปที่ 6.13 แสดงผลการสังเคราะห์วงจรดีฟลิปฟลอปโดยใช้ฟลิปฟลอปดีสร้างโดยตรง

2. การสร้างวงจรบล็อกในรูปแบบต่างๆ เช่นการผสมวงจรบล็อกเพื่อสร้างวงจรบล็อกชนิดใหม่ขึ้น โดยจะนำตัวอย่างการสร้างวงจรใหม่จากวงจรบล็อกตัวอย่างดีฟลิปฟลอปและวงจรบล็อกตัวอย่างเจเคฟลิปฟลอป ดังนี้

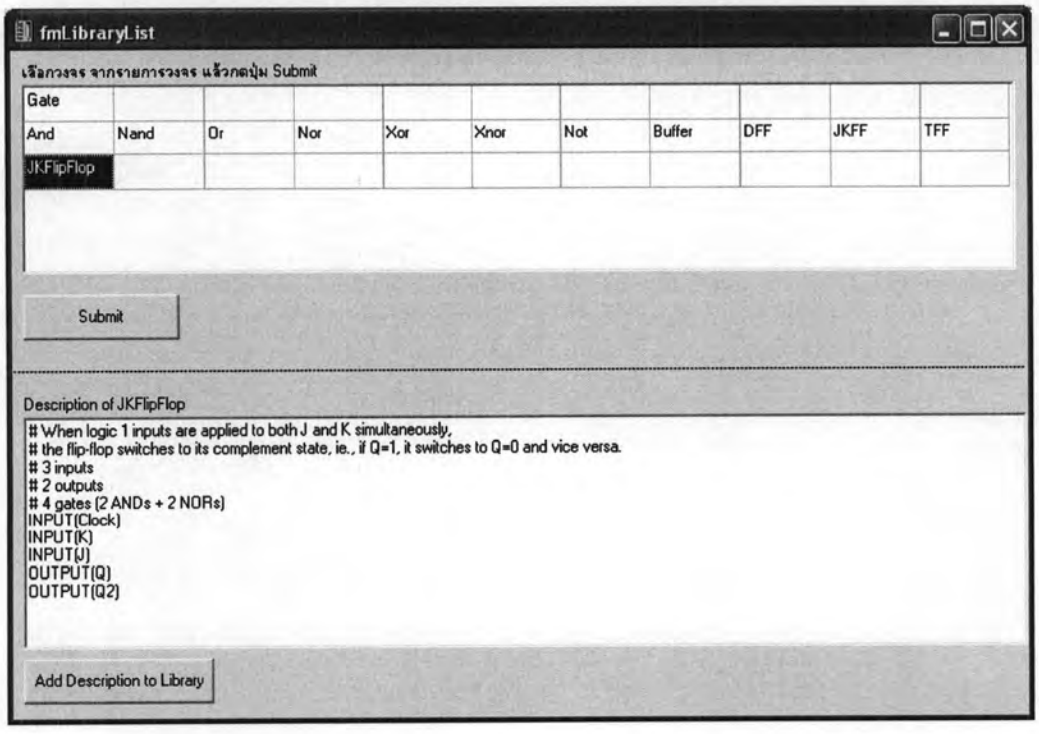
2.1. คลิกเลือกปุ่ม ☐ เพื่อเลือกวงจรบล็อกตัวอย่างดีฟลิปฟลอปและวงจรบล็อกตัวอย่างเจเคฟลิปฟลอปและวาดลงในส่วนกราฟิก ดังจะแสดงในรูปที่ 6.14-6.17



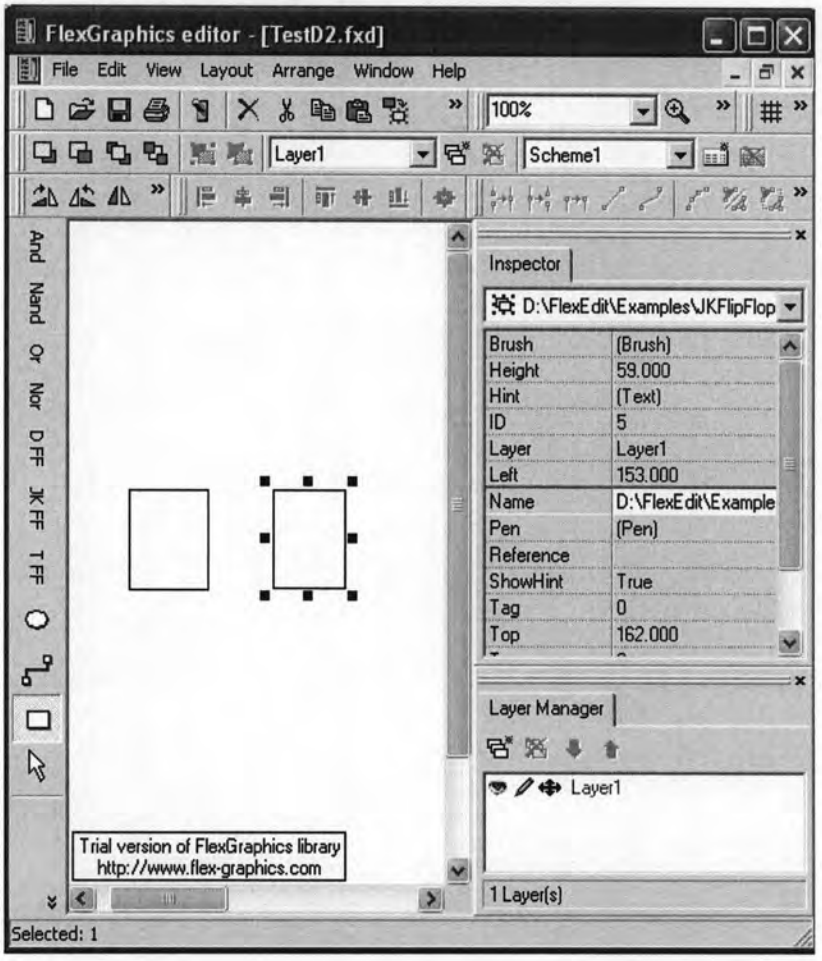
รูปที่ 6.14 แสดงการเลือกวงจรบล็อกตัวอย่างดีฟลิปฟลอปเพื่อวาดลงในส่วนกราฟิก



รูปที่ 6.15 แสดงการวาดวงจรบล็อกตัวอย่างดีฟลิปฟลอปลงในส่วนกราฟิก



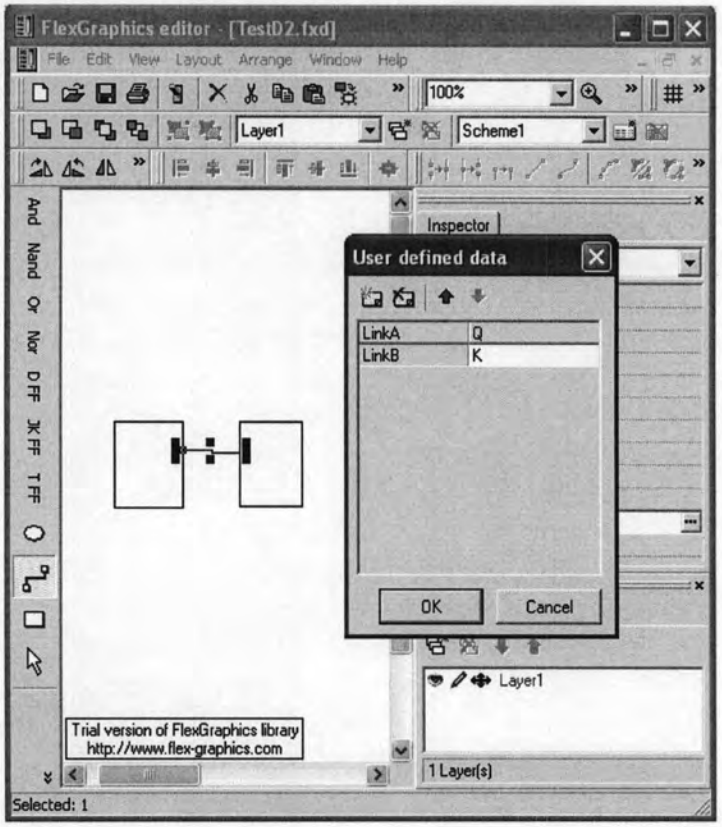
รูปที่ 6.16 แสดงการเลือกวงจรบล็อกตัวอย่างเจเคฟลิปฟลอปเพื่อวาดลงในส่วนกราฟิก



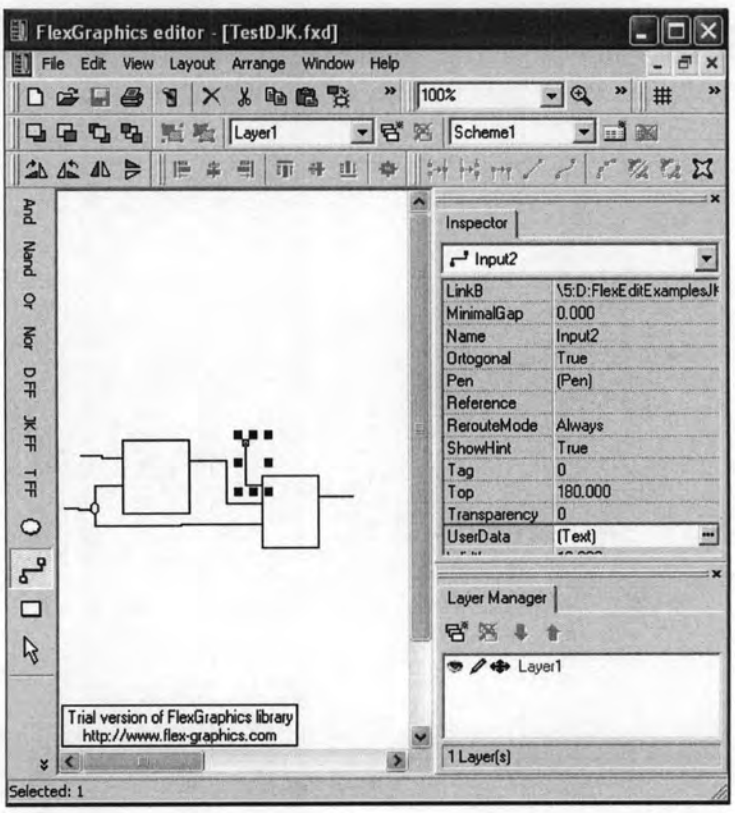
รูปที่ 6.17 แสดงการวาดวงจรบล็อกตัวอย่างเจเคฟลิปฟลอปลงในส่วนกราฟิก

2.2. ทดสอบการเชื่อมต่อสายสัญญาณระหว่าง 2 วงจรบล็อกตัวอย่างดีฟลิปฟลอปและวงจรบล็อกตัวอย่างเจเคฟลิปฟลอป และกำหนดการเชื่อมต่อระหว่างสายสัญญาณและสายสัญญาณเดิมของวงจรบล็อกตัวอย่างทั้งสอง ดังในรูปที่ 6.18

2.3 เพิ่มการเชื่อมต่ออื่นๆ จนครบทั้งวงจร ดังแสดงในรูปที่ 6.19 และแสดงผลการสังเคราะห์วงจรในรูปที่ 6.20



รูปที่ 6.18 แสดงการกำหนดการเชื่อมต่อของสายสัญญาณระหว่างวงจรบล็อก

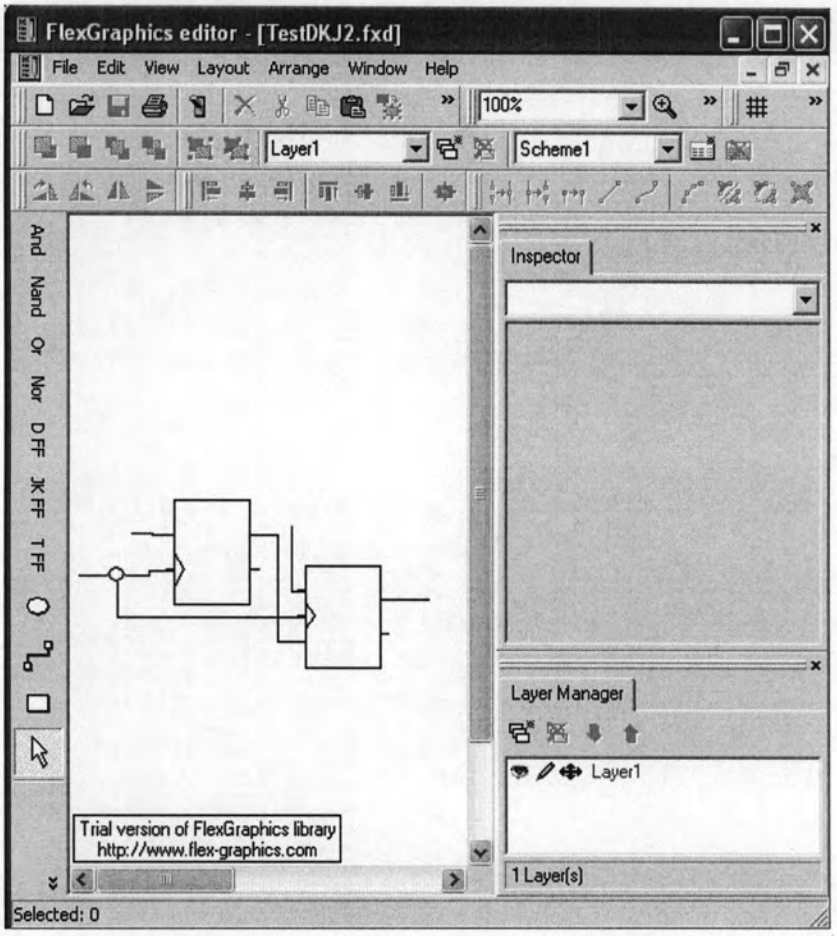


รูปที่ 6.19 แสดงการเชื่อมต่อสายสัญญาณอื่นๆ ของวงจรบล็อกตัวอย่าง

```
1   # 3 inputs
2   # 4 outputs
3   # 1 inverters
4   # 8 gates (2 ANDs + 2 NANDs + 4 NORs)
5
6   INPUT(CLK)
7   INPUT(InputD)
8   INPUT(InputJ)
9
10  OUTPUT(OutputJK)
11  OUTPUT(BLOCK4_Q2)
12  OUTPUT(BLOCK5_Q)
13  OUTPUT(BLOCK5_Q2)
14
15  BLOCK4_W1 = NOT(InputD)
16
17  BLOCK5_W4 = AND(CLK, BLOCK5_Q, Connector)
18  BLOCK5_W5 = AND(CLK, BLOCK5_Q2, InputJ)
19
20  BLOCK4_W2 = NAND(InputD, CLK)
21  BLOCK4_W3 = NAND(CLK, BLOCK4_W1)
22
23  Connector = NOR(BLOCK4_W2, BLOCK4_Q2)
24  BLOCK4_Q2 = NOR(BLOCK4_W3, Connector)
25  BLOCK5_Q = NOR(BLOCK5_W4, BLOCK5_Q2)
26  BLOCK5_Q2 = NOR(BLOCK5_W5, BLOCK5_Q)
```

รูปที่ 6.20 แสดงผลการสังเคราะห์วงจรจากวงจรบล็อกตัวอย่าง

2.4 สร้างการเชื่อมต่อรูปแบบเดียวกัน แต่ใช้อุปกรณ์ฟลิปฟลอปดีและเจเคแทนวงจรบล็อก เพื่อทดสอบว่าสามารถให้ผลการสังเคราะห์วงจรได้เหมือนกันหรือไม่ ดังรูปที่ 6.21 และ 6.22

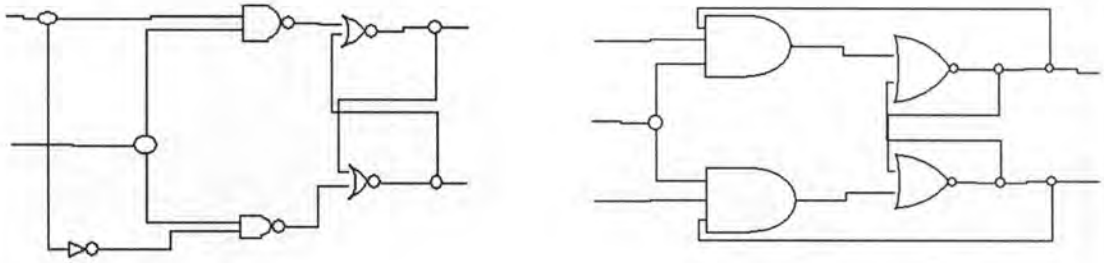


รูปที่ 6.21 แสดงการสร้างวงจรจากอุปกรณ์ฟลิปฟลอปดีและเจเค

```
1   # 3 inputs
2   # 1 outputs
3   # 1 D-type flipflops
4   # 1 JK-type flipflops
5
6   INPUT(InputD)
7   INPUT(CLK)
8   INPUT(InputJ)
9
10  OUTPUT(OutputJK)
11
12  Connector = DFF(InputD, CLK)
13
14  OutputJK = JKFF(Connector, CLK, InputJ)
15
```

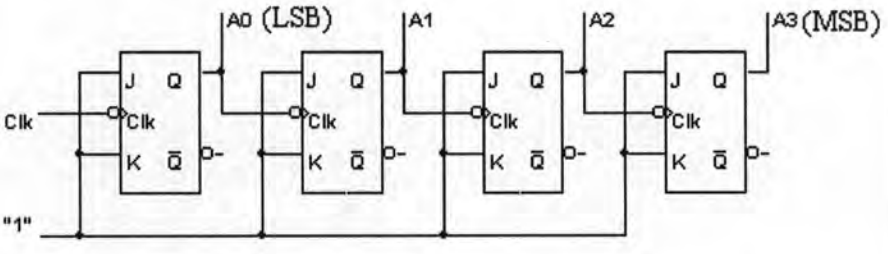
รูปที่ 6.22 แสดงผลการสังเคราะห์วงจรที่สร้างจากอุปกรณ์ฟลิปฟลอปดีและเจเค

2.5 จากการสังเคราะห์วงจรตัวอย่างทั้งสองรูปแบบในข้อ 2 นี้ จะพบว่าทั้งสองวงจรสามารถให้การสังเคราะห์วงจรได้ถูกต้องและเหมือนกัน (ตามในกรอบสีแดง) ซึ่งในการสร้างโดยใช้วิธีนี้พบว่าใช้แรงงานในการสร้างวงจรเพียงแค่ 10 ครั้ง ในขณะที่การสร้างวงจรเดียวกันโดยใช้วิธีปกติซึ่งก็คือการสร้างวงจรดีและวงจรเจเคใหม่ให้มาเชื่อมต่อกันต้องใช้แรงงานในการสร้างวงจรประมาณ 50 ครั้ง ดังรูปที่ 6.23 แสดงการสร้างวงจรดีและวงจรเจเค



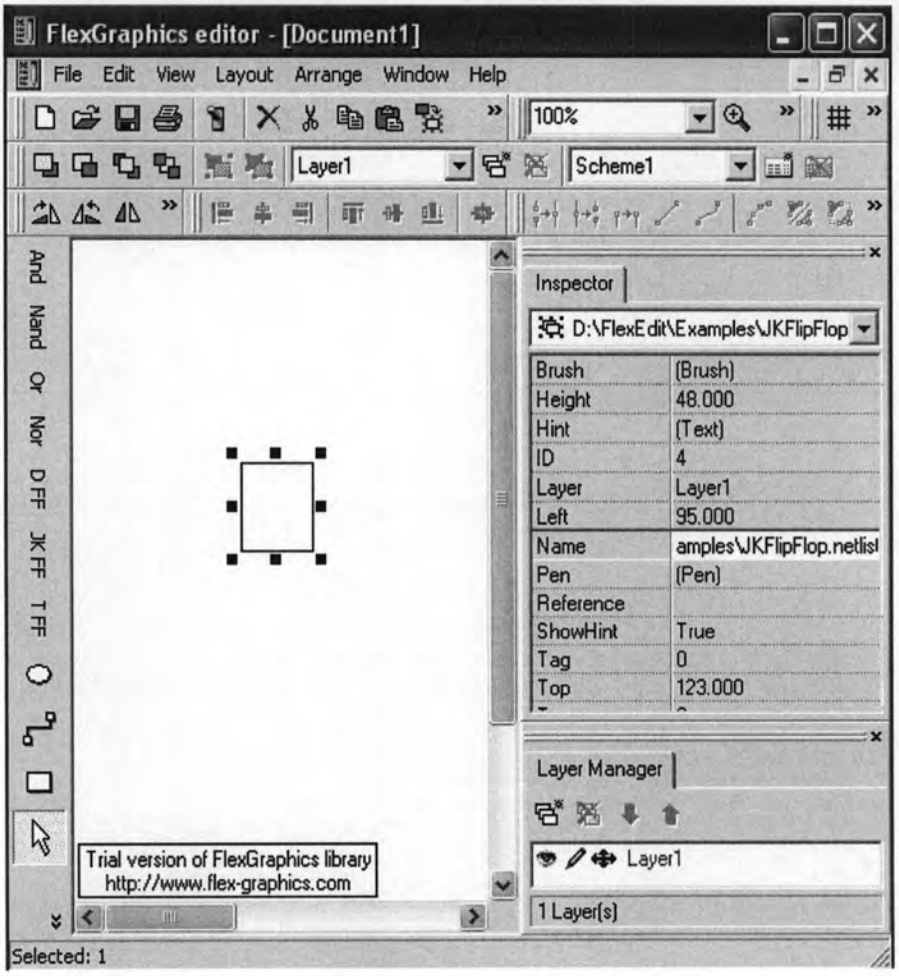
รูปที่ 6.23 แสดงการสร้างวงจรดีและวงจรเจเคด้วยวิธีปกติ

3. การสร้างวงจรบล็อกที่มีจำนวนวงจรภายในวงจรบล็อกมากกว่าหนึ่งวงจร หรือวงจรกลุ่มหรือวงจรหลายบิท ในข้อนี้จะนำตัวอย่างวงจรนับ 4 บิท ที่สร้างโดยเจเคฟลิปฟลอป และจะสร้างตัวอย่างวงจรนับที่สร้างโดยวงจรบล็อกตัวอย่างเจเคฟลิปฟลอป วงจรนับมีลักษณะดังแสดงในรูปที่ 6.24 และทำการทดสอบวงจรนับเจเค 4 บิทดังนี้



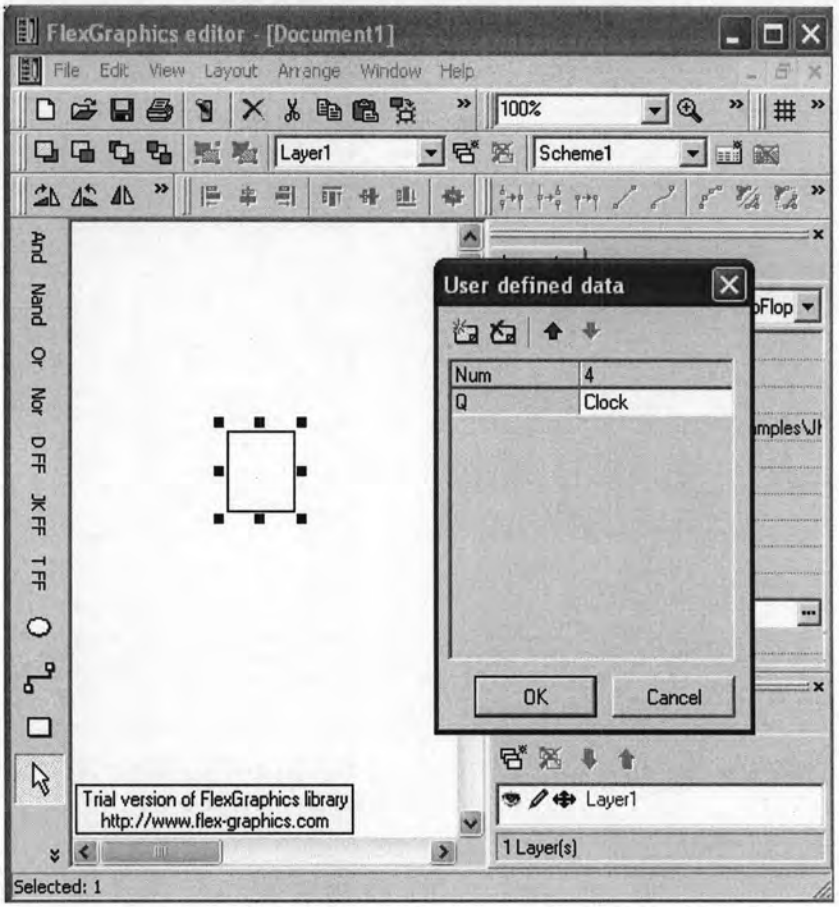
รูปที่ 6.24 แสดงวงจรนับ 4 บิทโดยใช้เจเคฟลิปฟลอป

3.1 เลือกวงจรบล็อกจากวงจรตัวอย่างเจเคฟลิปฟลอปแล้วนำมาวาดลงเครื่องมือส่วนกราฟิกดังรูปที่ 6.25



รูปที่ 6.25 แสดงการเพิ่มวงจรบล็อกลงในวงจร

3.2 คลิกคุณสมบัติ UserData ที่ช่อง Inspector ด้านขวามือของเครื่องมือจะขึ้นกล่องโต้ตอบให้เพิ่มข้อมูลจำนวนวงจรและการเชื่อมต่อระหว่างวงจรภายใน ดังแสดงในรูปที่ 6.26



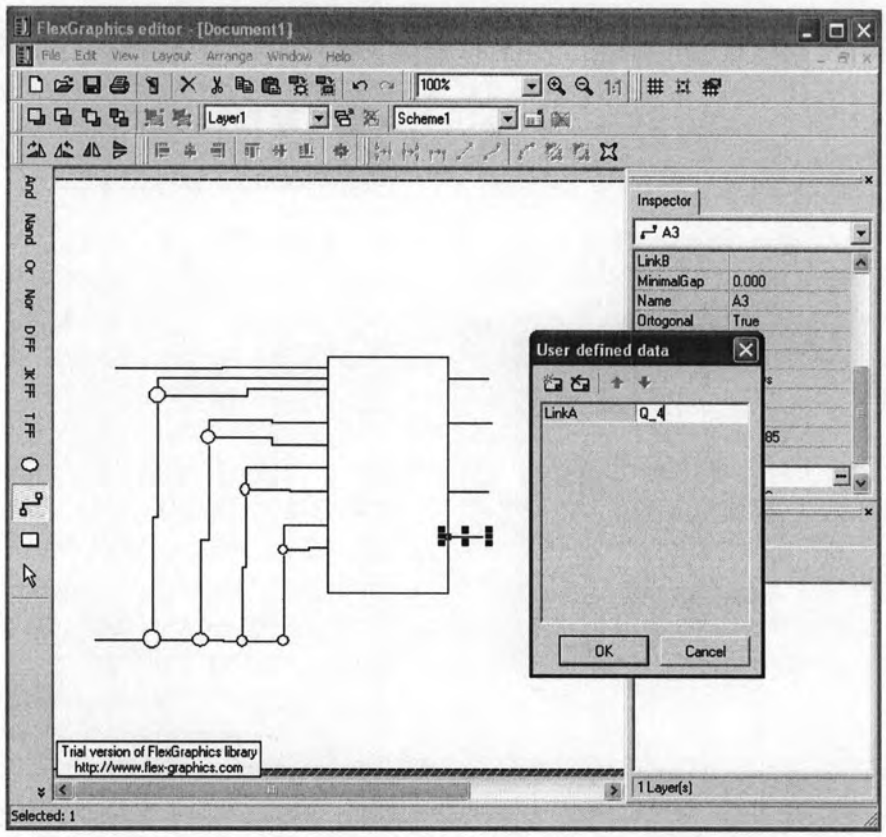
รูปที่ 6.26 แสดงการกำหนดจำนวนวงจรบล็อกภายในวงจรนับ และการเชื่อมต่อระหว่างวงจรบล็อกภายในวงจรนับ

3.3 เพิ่มการเชื่อมต่อสายสัญญาณจากภายนอกวงจรบล็อกมายังสายสัญญาณอินพุตภายในวงจรบล็อก

3.4 เพิ่มการเชื่อมต่อสายสัญญาณเอาต์พุตภายในวงจรบล็อกไปยังสายสัญญาณภายนอกวงจรบล็อกดังรูปที่ 6.27

3.5 ผลจากการสังเคราะห์วงจรตัวอย่างดังกล่าว จะแสดงในรูปที่ 6.28

3.6 เมื่อเทียบกับรูปที่ 6.24 แล้วจะพบว่า นอกจากจะประหยัดแรงงานในการสร้างวงจรทุกวงจรบล็อกแล้ว ยังประหยัดแรงงานโดยการกำหนดการต่อกันระหว่างบิทเพียงแค่ครั้งเดียว ไม่ต้องเชื่อมต่อกันระหว่างบิททุกบิทอีกด้วย

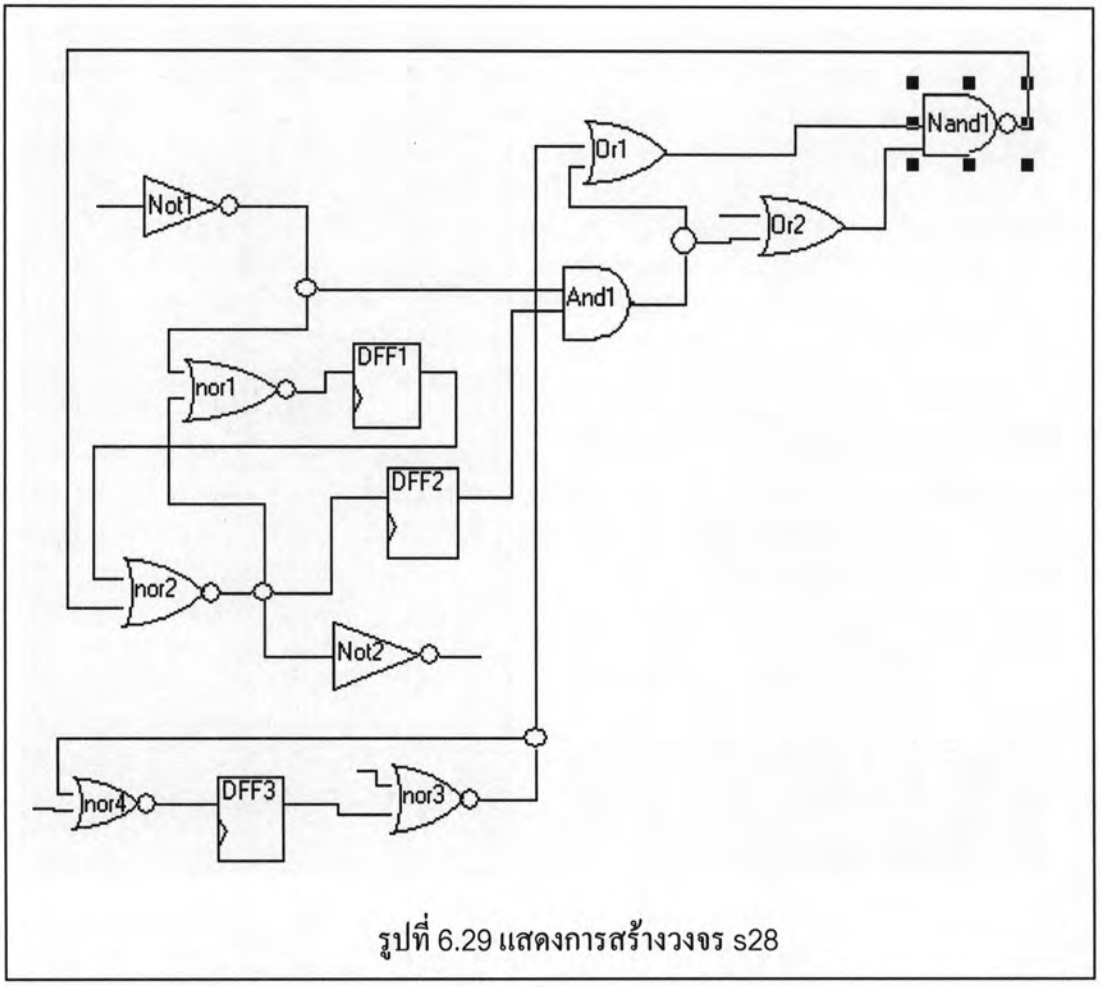


รูปที่ 6.27 แสดงการเชื่อมต่อระหว่างสายสัญญาณเอาต์พุตภายในวงจรนับไปยังสายสัญญาณภายนอก

```
5   INPUT(Data1)
6   INPUT(CLK)
7
8   OUTPUT(A0)
9   OUTPUT(A1)
10  OUTPUT(A2)
11  OUTPUT(A3)
12  OUTPUT(BLOCK4_Q2_1)
13  OUTPUT(BLOCK4_Q2_2)
14  OUTPUT(BLOCK4_Q2_3)
15  OUTPUT(BLOCK4_Q2_4)
16
17  BLOCK4_W4_1 = AND(CLK, A0, Data1)
18  BLOCK4_W4_2 = AND(A0, A1, Data1)
19  BLOCK4_W4_3 = AND(A1, A2, Data1)
20  BLOCK4_W4_4 = AND(A2, A3, Data1)
21  BLOCK4_W5_1 = AND(CLK, BLOCK4_Q2_1, Data1)
22  BLOCK4_W5_2 = AND(A0, BLOCK4_Q2_2, Data1)
23  BLOCK4_W5_3 = AND(A1, BLOCK4_Q2_3, Data1)
24  BLOCK4_W5_4 = AND(A2, BLOCK4_Q2_4, Data1)
25
26  A0 = NOR(BLOCK4_W4_1, BLOCK4_Q2_1)
27  A1 = NOR(BLOCK4_W4_2, BLOCK4_Q2_2)
28  A2 = NOR(BLOCK4_W4_3, BLOCK4_Q2_3)
29  A3 = NOR(BLOCK4_W4_4, BLOCK4_Q2_4)
30  BLOCK4_Q2_1 = NOR(BLOCK4_W5_1, A0)
31  BLOCK4_Q2_2 = NOR(BLOCK4_W5_2, A1)
32  BLOCK4_Q2_3 = NOR(BLOCK4_W5_3, A2)
33  BLOCK4_Q2_4 = NOR(BLOCK4_W5_4, A3)
34
```

รูปที่ 6.28 แสดงผลการสังเคราะห์วงจรนับที่สร้างขึ้นจากวงจรบล็อกเจเคฟลิปฟลอป

4. ตัวอย่างการสร้างวงจรจากวงจรมาตรฐาน ISCAS89 1 วงจรได้แก่ วงจร s27 ดังรูปที่ 6.29 และผลการสังเคราะห์ของ s27.netlist และวงจร s27.bench จากวงจรมาตรฐาน ISCAS89 พบว่าตรงกัน ตัวอย่างอื่นๆ ของวงจรมาตรฐาน ISCAS89 วงจรอื่นๆ ซึ่งมีจำนวนอุปกรณ์จำนวนมากไม่ได้นำมาทดสอบ



รูปที่ 6.29 แสดงการสร้างวงจร s28

จากการทดลองสร้างวงจรในรูปแบบต่างๆ แล้ว พบว่าสามารถให้ผลการสังเคราะห์วงจรที่ถูกต้อง ประหยัดแรงงานในการออกแบบวงจรและรูปแบบวงจรดังกล่าวนี้ยังครอบคลุมการสร้างวงจรในรูปแบบอื่นๆ ต่อไป

ประสิทธิภาพของการทำงาน

เมื่อวัดเวลาที่ใช้ในการสังเคราะห์วงจรและการจำลองการทำงาน ในขณะที่จำนวนบิตของวงจรบล็อกมากขึ้นแล้ว พบว่าเมื่อมีจำนวนบิตมากขึ้นก็ใช้เวลาในการสังเคราะห์วงจรและจำลองการทำงานนานมากขึ้น แต่ยังใช้เวลาน้อยกว่างานวิจัยที่ใช้แนวคิดเชิงวัตถุในการออกแบบวงจรตรรกะเชิงผสม ดังแสดงผลในตารางที่ 2 แต่อย่างไรก็ตาม เวลาที่ใช้ในงานวิจัยนี้และในงานวิจัยก่อนหน้า ไม่สามารถเปรียบเทียบกันได้โดยตรง เนื่องจากสภาพแวดล้อมในการทดลองไม่เท่ากัน

ตารางที่ 2 แสดงความสัมพันธ์ระหว่างจำนวนบิท(จำนวนเกท)และเวลาที่ใช้ในการสังเคราะห์วงจรและจำลองการทำงานเป็นวินาทีและนาที ตามลำดับ และเทียบกับเวลาที่ใช้ในงานวิจัยก่อนหน้า

จำนวนบิท (เทียบเป็นจำนวนเกท)	สังเคราะห์วงจร (วินาที)		จำลองการทำงาน (นาที)	
	งานวิจัยก่อน	งานวิจัยนี้	งานวิจัยก่อน	งานวิจัยนี้
100 (500)	8	น้อยกว่า 1	85 ชั่วโมง	5
200(1000)	10	น้อยกว่า 1	288 ชั่วโมง	18.67
300 (1500)		น้อยกว่า 1		38.58
400 (2000)		น้อยกว่า 1		66.25
500 (2500)		น้อยกว่า 1		94.32
1000 (5000)		1		365.93

เมื่อเปรียบเทียบความสามารถในการทำงาน ระหว่างเครื่องมือจำลองการทำงานชนิดอื่น เครื่องมือออกแบบวงจรตรรกะเชิงผสม และเครื่องมือออกแบบวงจรตรรกะเชิงลำดับแล้ว จะแสดงการเปรียบเทียบได้ดังตารางที่ 3

ตารางที่ 3 แสดงการเปรียบเทียบความสามารถในการทำงานระหว่างเครื่องมือจำลองการทำงาน เครื่องมือออกแบบวงจรตรรกะเชิงผสม และเครื่องมือออกแบบวงจรตรรกะเชิงลำดับ

คุณสมบัติ	LogicWorks	เครื่องมือออกแบบวงจรตรรกะเชิงผสม	เครื่องมือออกแบบวงจรตรรกะเชิงลำดับ
การใช้งานแบบกราฟิก	มีการเลือกใช้เกทหลายชนิดหลายครั้ง	มีการเรียกใช้งานหลายขั้นตอน	ปรับปรุงให้มีการเลือกเกทครั้งเดียว และมีความยืดหยุ่นในการออกแบบมากขึ้น
การสังเคราะห์วงจร	ไม่สามารถทำได้	สามารถทำได้ดี	สามารถทำได้ดีและใช้เวลาลดลง
จำลองการทำงาน	สามารถทำได้ดี	สามารถทำได้ แต่ใช้เวลามาก	เชื่อมต่อกับเครื่องมือจำลองการทำงานชนิดอื่น