บทที่ 3

การพัฒนาและปรับปรุงโปรแกรมคอมพิวเตอร์

3.1 ชุดโปรแกรม VPI

ชุดโปรแกรม VPI เป็นชุดโปรแกรมที่ใช้เพื่อการคำนวณการจัดการเชื้อเพลิงในเครื่องปฏิกรณ์นิวเคลียร์ พัฒนาขึ้นครั้งแรกที่ Virginia Polytechnical Institute and State University สำหรับระบบคอมพิวเตอร์ไอบีเอ็ม (IBM main frame) และได้ทำการดัดแปลงให้ใช้ได้กับระบบคอมพิวเตอร์ของซีดีซี (CDC) ที่ Institute of Nuclear Research, Swierk, Poland จุดมุ่งหมายหลักของชุดโปรแกรม VPI คือเพื่อการเรียนการสอน (education purpose) ในด้านการจัดการเชื้อเพลิงในเครื่องปฏิกรณ์โดยเฉพาะแบบ PWR การคำนวณใช้แบบจำลองที่ทำให้ง่ายลงอย่างมาก จึงทำให้ผลการคำนวณไม่ถูกต้องเพียงพอที่จะใช้งานกับเครื่องปฏิกรณ์จริง ชุดโปรแกรมประกอบด้วย 6 มอดูล แต่ละมอดูลทำงานอย่างอิสระไม่ขึ้นต่อกันแต่ผลลัพธ์ที่ได้จากมอดูลหนึ่งจะมีประโยชน์ในการเตรียมอินพุทของมอดูลอื่น

ชุดโปรแกรม VPI ประกอบด้วยมอดูลต่างๆ ดังต่อไปนี้

(1) FARCON เป็นมอดูลที่ใช้คำนวณค่าคงที่ต่างๆ สำหรับกลุ่มนิวตรอนเร็วและเรโซแนนซ์ (fast and resonance group constants)

(2) SLOCON เป็นมอดูลที่ใช้คำนวณค่าคงที่ต่างๆ สำหรับกลุ่มนิวตรอนเทอร์มัล (thermal group constants)

(3) DISFAC เป็นมอดูลที่ใช้คำนวณค่าเทอร์มัลยูทิไลเซชันแฟกเตอร์ (thermal utilization factor) และค่าดิสแอดวานเตจแฟกเตอร์ (disadvantage factor)

(4) ODOG เป็นมอดูลที่ใช้คำนวณสมการการแพร่กระจายของนิวตรอนแบบหนึ่งกลุ่มหนึ่งมิติ (one dimension one group diffusion equation)

(5) ODMUG เป็นมอดูลที่ใช้คำนวณสมการการแพร่กระจายของนิวตรอนแบบสามกลุ่มหนึ่งมิติและการคำนวณสภาวะวิกฤต (one dimension three group diffusion equation and criticality calculation)

(6) FUELBURN เป็นมอดูลที่ใช้คำนวณเบิร์นอัพหรือการใช้เชื้อเพลิงของเครื่องปฏิกรณ์ (burnup or depletion calculation)

ชุดโปรแกรม VPI ทั้ง 6 มอดูลนี้เขียนด้วยภาษาฟอร์แทรน-IV และสามารถใช้คำนวณได้ด้วยเครื่องมินิคอมพิวเตอร์ PRIME 9750 ของศูนย์คอมพิวเตอร์คณะวิศวกรรมศาสตร์ โดยใช้เวลาในการคำนวณไม่มากนัก

3.2 การปรับปรุงชุดโปรแกรม VPI เพื่อใช้กับไมโครคอมพิวเตอร์ไอบีเอ็มพีซี

ในการปรับปรุงชุดโปรแกรม VPI เพื่อใช้กับไมโครคอมพิวเตอร์ ประกอบด้วยส่วนที่สำคัญดังต่อไปนี้

3.2.1 การย้ายชุดโปรแกรมจากเทปแม่เหล็ก (magnetic tape)

เป็นการย้ายไฟล์ (file) ชนิดต่างๆ ของชุดโปรแกรม VPI จากเทปแม่เหล็กซึ่งเป็นแหล่งเก็บข้อมูลอันหนึ่งของระบบคอมพิวเตอร์ขนาดใหญ่ไปเก็บไว้ในแผ่นดิสเก็ตต์ (diskette) ซึ่งเป็นแหล่งเก็บข้อมูลสำหรับไมโครคอมพิวเตอร์ไอบีเอ็มพีซี สาเหตุที่ต้องมีการย้ายไฟล์เนื่องจากในไอบีเอ็มพีซีไม่มีระบบขับเคลื่อนเทปแม่เหล็ก (magnetic tape drive) ในการย้ายไฟล์ได้ใช้ระบบมินิคอมพิวเตอร์ PRIME 9750 ของศูนย์คอมพิวเตอร์คณะวิศวกรรมศาสตร์เป็นตัวกลาง โดยขั้นตอนต่างๆ แสดงเรียงตามลำดับดังนี้

นำเทปแม่เหล็กที่เก็บชุดโปรแกรม VPI ไปติดเข้ากับระบบขับเทปแล้วใช้เทอร์มินัลของ PRIME โดยใช้คำสั่งดังนี้ (คำที่ขีดเส้นใต้เป็นคำที่ต้องพิมพ์)

คำสั่ง	คำอธิบาย
OK, AS MTØ	ติดต่อตู้ขับเทปแม่เหล็ก unit Ø
OK, MAGNET	ขอใช้คำสั่งที่เกี่ยวข้องกับเทปแม่เหล็ก
> R	อ่านไฟล์จากเทปแม่เหล็กสู่จานแม่เหล็ก (magnetic disk)
MT # = Ø	ตู้ขับเทปแม่เหล็กยูนิตที่ Ø
FILE # = 1	ไฟล์ที่ 1 ของเทปแม่เหล็ก
.	
.	
.	
DISK FILE = F1	ชื่อไฟล์ F1 ที่เก็บในจานแม่เหล็ก
> R	ทำการอ่านไฟล์ใหม่

ทำการย้ายไฟล์ของชุดโปรแกรม VPI ทั้งหมดไปเก็บในจานแม่เหล็กของระบบ PRIME หลังจากนั้นใช้ไอบีเอ็มพีซีที่เชื่อมโยงอยู่กับระบบ PRIME ทำการเคลื่อนย้ายข้อมูลจากจานแม่เหล็กของระบบ PRIME เข้าไปเก็บไว้ในดิสเก็ตต์ของไอบีเอ็มพีซีโดยใช้โปรแกรม PRIMELINK ซึ่งเป็นโปรแกรมรับส่งข้อมูลระหว่างระบบ PRIME กับ ไอบีเอ็มพีซี

3.2.2 การใช้โปรแกรม PRIMELINK ด้วยเครื่องไอบีเอ็มพีซี

ใส่แผ่น PRIMELINK system #1 ใน drive A และ system #2 ใน drive B ของเครื่องไอบีเอ็มพีซีที่เชื่อมกับระบบ PRIME พิมพ์

A> PRIMELINK

เลือกโมด emulation ทำให้ไอบีเอ็มพีซีเป็นเสมือนเทอร์มินัลตัวหนึ่งของระบบ PRIME โดยทำการ LOGIN พิมพ์ USER ID และ PASS WORD เสร็จแล้วพิมพ์ "PRIMELINK" กลับไปที่เมนู เลือกโมด transfer file เลือกโมด receive file เปลี่ยนแผ่น system #2 ใน drive B เป็นแผ่นที่จะใช้เก็บไฟล์ พิมพ์ชื่อไฟล์ที่จะเคลื่อนย้าย

host file name : F1 <return>

name : B:FARCON.FOR <return>

เมื่อการเคลื่อนย้ายไฟล์เสร็จเรียบร้อย เปลี่ยนแผ่นดิสก์ใน drive B กลับไปเป็น system #2 กลับไปที่เมนู เลือกโมด emulation ทำการ LOGOUT เป็นการสิ้นสุดการติดต่อกับระบบ PRIME

3.2.3 การดัดแปลงชุดโปรแกรม VPI

เนื่องจากชุดโปรแกรม VPI เดิมเขียนด้วยภาษาฟอร์แทรน-IV ซึ่งเป็นภาษาที่ใช้ในระบบคอมพิวเตอร์ขนาดกลางและขนาดใหญ่ แต่ไม่มีในเครื่องไอบีเอ็มพีซี อย่างไรก็ตาม ไอบีเอ็มพีซีก็มีภาษาที่มีรูปแบบใกล้เคียงกันมากคือภาษาฟอร์แทรน-77 (Microsoft FORTRAN-77) ดังนั้นจึงได้ดัดแปลงโปรแกรมจากภาษาฟอร์แทรน-IV ไปเป็นภาษาฟอร์แทรน-77 เพื่อให้สามารถคอมไพล์ (compile) และประมวลผล (execute) ได้ด้วยไอบีเอ็มพีซี ส่วนที่ได้มีการดัดแปลงในชุดโปรแกรม VPI ที่สำคัญ ได้แก่

(1) การประกาศตัวแปรอักขระ (declaration of character variable) ในภาษาฟอร์แทรน-IV ไม่มีคำสั่งในการประกาศตัวแปรอักขระ แต่สำหรับฟอร์แทรน-77 จำเป็นต้องใช้คำสั่งเฉพาะสำหรับการประกาศตัวแปรอักขระ เช่น

```
CHARACTER*4 NAME(28),PLUS *1
CHARACTER*80 TITLE
```

เป็นการกำหนดตัวแปร NAME มี 28 ตัว ตั้งแต่ NAME(1) ถึง NAME(28) โดยตัวแปรแต่ละตัวประกอบด้วยอักขระ 4 ตัว กำหนดตัวแปร PLUS เป็นตัวแปรอักขระมีความยาว 1 ตัวอักษร และกำหนดตัวแปร TITLE มีความยาว 80 ตัวอักษร

(2) การควบคุมการปิด-เปิดไฟล์ในระหว่างการประมวลผล ในชุดโปรแกรมเดิมที่ใช้กับคอมพิวเตอร์ขนาดใหญ่และมินิคอมพิวเตอร์ การติดต่อระหว่างอุปกรณ์ต่าง ๆ เช่นจานแม่เหล็ก เทปแม่เหล็ก เครื่องพิมพ์ (line printer) และอื่น ๆ ทำได้โดยการใช้ JCL (job control langauge) เพื่อแจ้งความต้องการไปให้ระบบจัดการ (operating system, OS) ทราบว่าในการประมวลผล ต้องการใช้ไฟล์ชื่ออะไร และอยู่ที่ไหน สำหรับในไอบีเอ็มพีซี เนื่องจากไม่สามารถใช้ JCL จึงได้ทำการดัดแปลงให้เป็นลักษณะการเรียกใช้ไฟล์ระหว่างการประมวลผล โดยเลือกใช้คำสั่ง OPEN และ CLOSE รูปแบบคำสั่งเป็นดังนี้

```
OPEN(unit number,FILE='filename',STATUS='NEW' or 'OLD',
ACCESS='SEQUENTIAL' or 'DIRECT',FORM='FORMATTED' or
'UNFORMATTED',RECL=record length in bytes)
```

ตัวอย่างเช่น

```
OPEN(5,FILE='FARCON.DAT',STATUS='OLD',
ACCESS='SEQUENTIAL',FORM='FORMATTED')
```

เป็นคำสั่งเปิดไฟล์ยูนิตที่ 5 ในดิสเก็ตต์ชื่อไฟล์ FARCON.DAT เป็นไฟล์เก่าชนิดซีเควนเชียล (sequential file) คำสั่ง OPEN ต้องใช้คู่กับคำสั่ง CLOSE ซึ่งมีรูปแบบคือ

```
CLOSE (5)
```

เป็นการปิดไฟล์ยูนิตที่ 5 คำสั่งทั้งสองจะปรากฏอยู่ในโปรแกรมในตำแหน่งที่เหมาะสม การเรียกใช้ไฟล์อื่นในระหว่างการประมวลผลก็จะเป็นไปอย่างอัตโนมัติ

(3) การสร้างไฟล์ไลบรารี (library) บนดิสเก็ตต์ มอดูล FARCON ต้องการข้อมูลไลบรารีที่เขียนเป็นไฟล์ชนิดไม่มีฟอร์แมท (unformatted file) เก็บอยู่ในเทปแม่เหล็ก และในระหว่างประมวลผลจะทำการขับเคลื่อนเทปแม่เหล็กเพื่ออ่านข้อมูลมาประมวลผลในมอดูล แต่เนื่องจากไอบีเอ็มพีซีไม่มีระบบขับเคลื่อนเทปแม่เหล็กจึงได้ดัดแปลงให้มีการเขียนไฟล์ไลบรารีบนดิสเก็ตต์ โปรแกรมที่ดัดแปลงให้เขียนไฟล์ไลบรารีมีชื่อว่า LIBCR.FOR ในระหว่างการประมวลผลโปรแกรมจะอ่านข้อมูลจากไฟล์ชื่อ LIBRARY ซึ่งเป็นไฟล์ชนิดมีฟอร์แมท (formatted file) แล้วทำการเขียนไฟล์ใหม่ชื่อ LIB.BIN ที่เป็นไฟล์ชนิดไม่มีฟอร์แมท และมีรูปแบบที่มอดูล FARCON สามารถอ่านได้
คำสั่งที่สำคัญในโปรแกรม LIBCR.FOR

```
      OPEN(5,FILE='LIBRARY',FORM='FORMATTED',STATUS='OLD')
      OPEN(6,FILE='LIB.BIN',FORM='UNFORMATTED',STATUS='NEW')
        .
        .
        .
      READ(5,10) A,B,C
      WRITE(6) A,B,C
        .
        .
        .
   10 FORMAT(3F10.3)
        .
        .
        .
      CLOSE (5)
      CLOSE(6)
      STOP
      END
```

(4) ความสามารถในการรับข้อมูลตัวเลข ในระบบคอมพิวเตอร์ขนาดใหญ่ สามารถประมวลผลกับตัวเลขจำนวนจริงที่มีขนาดน้อยถึงระดับ 10^{-70} ในแบบความเที่ยงตรงปกติ (single precision) แต่สำหรับไอบีเอ็มพีซีทำได้เพียงระดับ 10^{-38} เท่านั้น ในการแก้ไขปรับปรุงมอดูล FUELBURN แม้ว่าจะสามารถ

คอมไพล์ได้โดยไม่มีข้อผิดพลาดเกิดขึ้น แต่เมื่อทำการประมวลผลเกิดความผิดพลาดชนิด "REAL devided by zero" ขึ้น จากการตรวจสอบพบว่ามีตัวแปรบางตัวที่มีค่าในลำดับ 10^{-40} ซึ่งเครื่องจะถือว่าเป็นศูนย์ เมื่อนำตัวแปรนี้ไปหารตัวแปรอื่นก็จะเกิดความผิดพลาดขึ้นดังกล่าว การแก้ไขได้ทำการเปลี่ยนแปลงการคำนวณในมอดูลจากแบบความเที่ยงตรงปกติไปเป็นแบบความเที่ยงตรงสองเท่า (double precision) ด้วยการกำหนดให้ตัวแปรทั้งหมดเป็นชนิดความเที่ยงตรงสองเท่า โดยใช้คำสั่ง

```
IMPLICIT DOUBLE PRECISION (A-H),(L-Z)
```

เป็นการกำหนดให้ ตัวแปรที่มีชื่อขึ้นต้นด้วย A ถึง Z ยกเว้น I,J,K เป็นตัวแปรชนิดความเที่ยงตรงสองเท่า นอกจากนี้ ฟังก์ชันต่างๆ ก็ต้องเปลี่ยนเป็นชนิดความเที่ยงตรงสองเท่าด้วย เช่น

ความเที่ยงตรงปกติ	ความเที่ยงตรงสองเท่า
ABS(X)	DABS(X)
EXP(X)	DEXP(X)

การใช้ความเที่ยงตรงสองเท่าในการคำนวณ แม้ว่าจะได้คำตอบที่ถูกต้อง แต่ก็ต้องใช้เวลาในการคำนวณนานกว่าในแบบความเที่ยงตรงปกติ ด้วยเหตุผลนี้จึงไม่ทำการเปลี่ยนแปลงมอดูลอื่นให้เป็นแบบความเที่ยงตรงสองเท่า เมื่อพบว่าคำตอบที่ได้จากแบบปกตินั้นใกล้เคียง หรือตรงกันกับคำตอบจากต้นฉบับเดิม

(5) การสร้างโปรแกรมย่อยฟังก์ชัน ERF(X) คือ error function ปกติจะเป็นฟังก์ชันไลบรารี (library function) ของระบบคอมพิวเตอร์ขนาดใหญ่ที่ใช้ในงานวิทยาศาสตร์และวิศวกรรม แต่สำหรับไอบีเอ็มพีซีไม่มีฟังก์ชันพิเศษอันนี้ มอดูล SLOCON ซึ่งเป็นมอดูลหนึ่งในชุดโปรแกรม VPI จำเป็นต้องใช้ฟังก์ชันนี้ในการประมวลผล จึงต้องทำการสร้างฟังก์ชัน ERF(X) ขึ้นโดยเขียนด้วยภาษาฟอร์แทรน-77 ในลักษณะโปรแกรมย่อยฟังก์ชัน (function subprogram) โดยมีรูปแบบที่มอดูล SLOCON สามารถเรียกใช้ได้ทันที

จากสมการ error function (5)

$$ERF(X) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt$$

เมื่อกระจายโดยใช้อนุกรมกำลัง (power series)

$$ERF(X) = \frac{2}{\sqrt{\pi}}\left(X - \frac{X^3}{3\cdot 1!} + \frac{X^5}{5\cdot 2!} + \frac{X^7}{7\cdot 3!} - \ldots \right)$$

ทำการเขียนโปรแกรมย่อยฟังก์ชัน ERF(X) ด้วยภาษาฟอร์แทรน-77 จากฟอร์มของอนุกรมกำลัง รายละเอียดของโปรแกรมมีดังนี้

```
      FUNCTION ERF(X)
      DIMENSION CC(2)
      PI=2.*X/SQRT(3.1415926)
      A=1.
      EN=0.
      CC(1)=1.
   10 B=-X*X*(2.*EN+1.)/((2.*EN+3.)*(EN+1.))*A
      CC(2)=CC(1)+B
      IF(ABS((CC(1)-CC(2))*PI).LT.1.E-8)GOTO 20
      A=B
      CC(1)=CC(2)
      EN=EN+1.
      GOTO 10
   20 ERF=CC(2)*PI
      RETURN
      END
```

การทดสอบความถูกต้องของโปรแกรม ใช้วิธีการตรวจสอบผลลัพธ์จากมอดูล SLOCON ถ้ามอดูล SLOCON เมื่อประมวลผลด้วยไอบีเอ็มพีซีโดยใช้โปรแกรมย่อยฟังก์ชัน ERF(X) ที่สร้างขึ้นใหม่ ให้ผลลัพธ์ตรงกับที่ได้จากการประมวลผลบนระบบคอมพิวเตอร์ขนาดใหญ่ ก็ถือว่าโปรแกรมย่อยฟังก์ชัน ERF(X) ใช้งานได้

(6) การปรับปรุงรูปแบบการอ่านข้อมูล โปรแกรมเดิมมีรูปแบบในการอ่านข้อมูลที่แตกต่างกัน เช่นรูปแบบ I (format I) สำหรับการอ่านข้อมูลจำนวนเต็ม รูปแบบ F (format F) สำหรับการอ่านข้อมูลเลขทศนิยม ทำให้ยุ่งยากกับผู้ใช้ที่ไม่มีความรู้ภาษาฟอร์แทรน โปรแกรมใหม่ได้เปลี่ยนแปลงรูปแบบการอ่านข้อมูลเป็นแบบอิสระ (free format or list directed I/O) ทำให้ง่ายต่อการเตรียมข้อมูล เช่น

แบบเดิม
```
READ(5,10) JJ,RADIUS,PITCH,NGP,BUCK
10 FORMAT(I3,2F10.4,I1,F10.7)
```

การเตรียมข้อมูล สำหรับค่า JJ=10, RADIUS=0.455, PITCH=1.834, NGP=3, BUCK=0.00331

```
คอลัมน์   12345678901234567890123456789012345678901234567890
          10 0.455          1.83430.00331
```

แบบใหม่
```
READ(5,*) JJ,RADIUS,PITCH,N,BUCK
```
การเตรียมข้อมูล
```
คอลัมน์   12345678901234567890123456789012345678901234567890
          10,.455,1.834,3,.00331
```

นอกจากการดัดแปลงดังที่ได้กล่าวแล้วซึ่งจัดเป็นส่วนสำคัญแล้ว ยังมีการแก้ไขประโยคคำสั่งบางอันและเปลี่ยนแปลงรูปแบบในการอ่าน-เขียนเสียใหม่ เพื่อให้เหมาะกับการแสดงผลบนจอและการพิมพ์ผลลัพธ์ออกทางเครื่องพิมพ์บนกระดาษขนาดกว้าง 9 นิ้ว

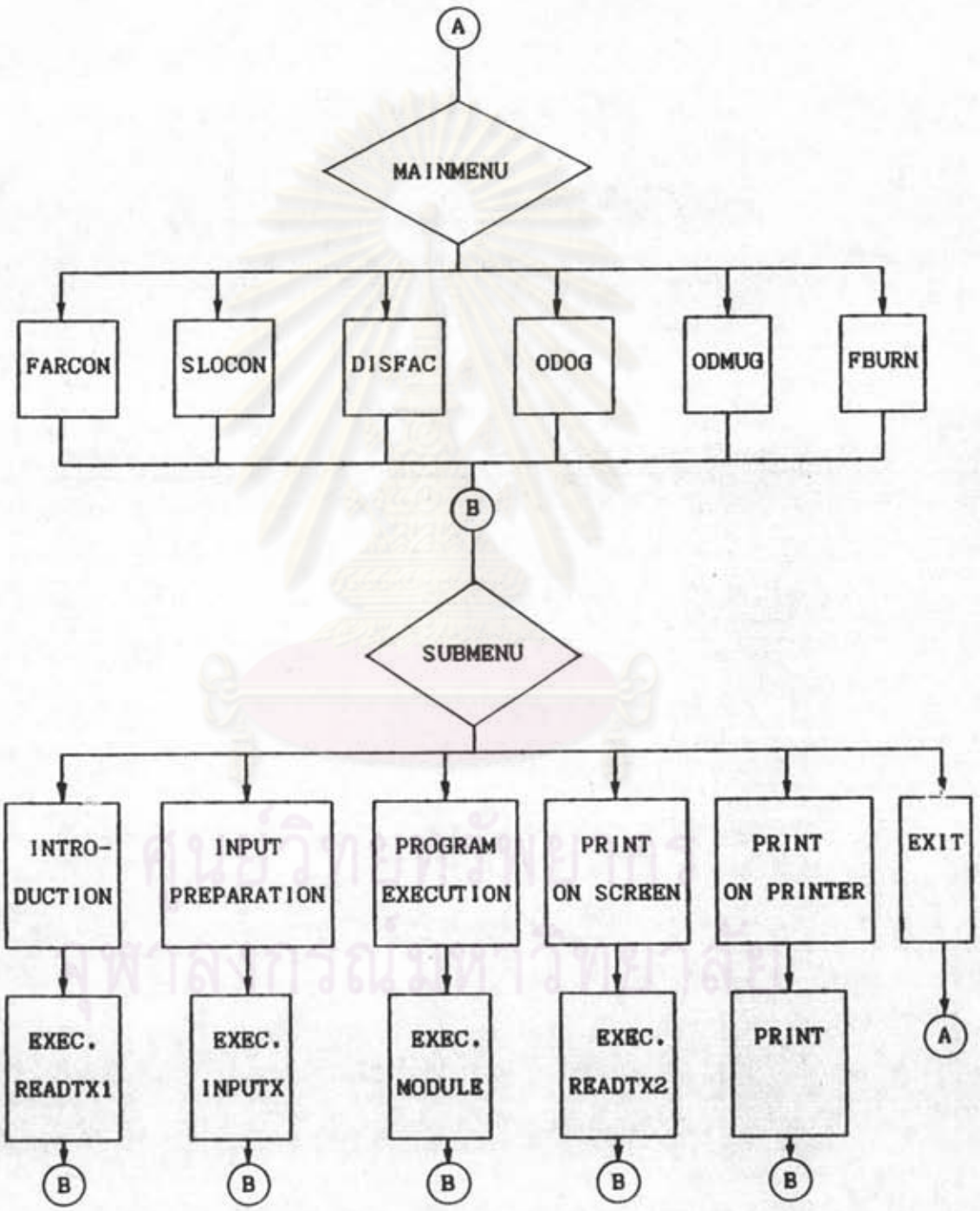
3.2.4 การพัฒนาโปรแกรมระบบจัดการ

โปรแกรมระบบจัดการในที่นี้หมายถึง โปรแกรมที่ทำหน้าที่เชื่อมโยงไฟล์ชนิดต่าง ๆ ให้เข้าอยู่ในระบบเดียวกัน ไฟล์ที่กล่าวถึงได้แก่ มอดูลต่าง ๆ ของชุดโปรแกรม VPI , โปรแกรมแสดงบทนำ ,โปรแกรมช่วยเตรียมข้อมูลอินพุท และโปรแกรมแสดงผลบนจอภาพ หน้าที่ต่างๆ ของโปรแกรมระบบจัดการ สรุปได้ดังนี้

(1) แสดงเมนูหลัก ให้ผู้ใช้เลือกว่าจะประมวลผลมอดูลใด
(2) แสดงเมนูย่อยของมอดูลที่ผู้ใช้เลือก จาก (1)
(3) แสดงคำอธิบายโปรแกรม และคำแนะนำการเตรียมข้อมูลอินพุท ของมอดูล
(4) แสดงข้อความบนจอภาพ เพื่อช่วยในการเตรียมข้อมูลอินพุท
(5) ส่งคำสั่งเพื่อทำการประมวลผลมอดูลตามที่ผู้ใช้เลือก
(6) แสดงผลเอาท์พุทของมอดูลที่ประมวลผลแล้วบนจอภาพ
(7) ส่งผลเอาท์พุทของมอดูลที่ประมวลผลแล้วไปพิมพ์ด้วยเครื่องพิมพ์

โปรแกรมระบบจัดการเขียนด้วยภาษาเบสิก โดยใช้ BASICA interpreter ประกอบด้วยโปรแกรมเชื่อมโยงหลัก 2 โปรแกรม คือโปรแกรม MAINMENU.BAS และ SUBMENU.BAS และมีโปรแกรมอื่นๆ อีก 8 โปรแกรม ทำหน้าที่แตกต่างกัน ขนาดของโปรแกรมรวมกันประมาณ 84000 ไบท์ (bytes) แผนภาพการ

ทำงานของโปรแกรม แสดงในรูป 3.1



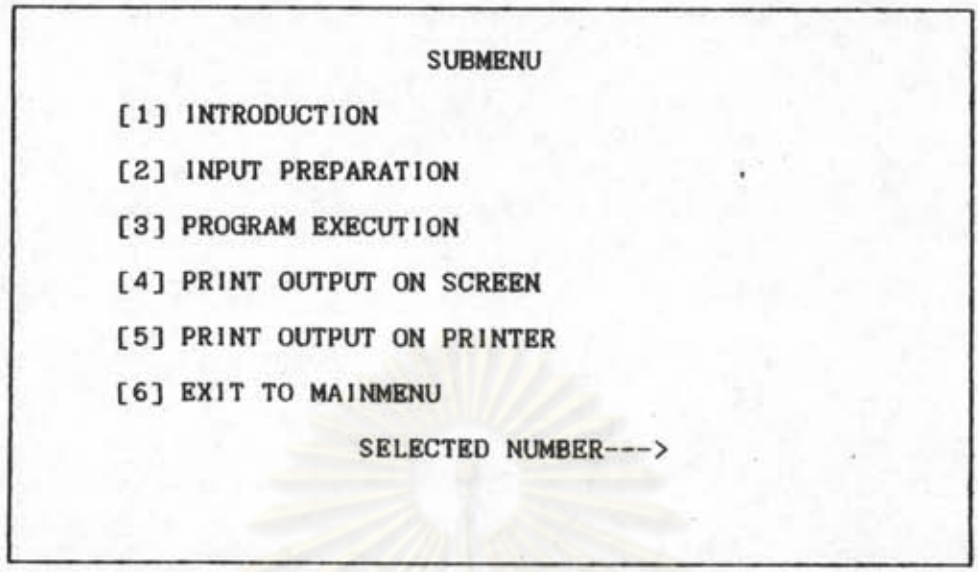
รูป 3.1 แผนภาพการทำงานของโปรแกรมระบบจัดการ

การทำงานของโปรแกรมเริ่มจาก โปรแกรมเมนูหลัก (MAINMENU) แสดงเมนูหลักให้ผู้ใช้เลือกมอดูลที่จะประมวลผล ดังแสดงในรูป 3.2 เมื่อผู้ใช้เลือกแล้วโปรแกรมเมนูหลักจะส่งค่าตัวแปรต่างๆ พร้อมทั้งส่งการทำงานไปให้กับโปรแกรมเมนูย่อย (SUBMENU) ทำการแสดงเมนูย่อยให้ผู้ใช้เลือกดังแสดงในรูป 3.3 สำหรับทางเลือกของผู้ใช้และลักษณะการจัดการในเมนูย่อย เป็นดังนี้

ทางเลือก	การจัดการของโปรแกรมเมนูย่อย
[1] Introduction	ส่งค่าตัวแปรและการทำงานให้โปรแกรม READTX1 ทำการอ่านไฟล์ที่เก็บคำอธิบายมอดูลมาแสดงบนจอ
[2] Input preparation	ส่งค่าตัวแปรและการทำงานให้กับโปรแกรมการเตรียมข้อมูลเพื่อช่วยการเตรียมข้อมูลอินพุท ของมอดูลที่เลือก
[3] execution	ทำการประมวลผลมอดูล
[4] print on screen	ส่งค่าตัวแปรและการทำงานให้กับโปรแกรม READTX2 ทำการอ่านไฟล์ที่เก็บเอาท์พุทของมอดูลที่ประมวลผลแล้ว
[5] print on printer	ส่งเอาท์พุทจากไฟล์ ไปพิมพ์ด้วยเครื่องพิมพ์
[6] exit	ส่งค่าตัวแปรและการทำงานให้กับโปรแกรมเมนูหลัก เพื่อให้ผู้ใช้เลือกประมวลผลมอดูลใหม่

```
                    MAINMENU
    [1] CALCULATE FAST AND RESONANCE GROUP CONSTANTS
    [2] CALCULATE THERMAL GROUP CONSTANTS
    [3] CALCULATE DISADVANTAGE FACTOR
    [4] SOLUTION OF 1-D 1-G DIFFUSION EQUATION
    [5] SOLUTION OF 1-D 3-G DIFFUSION EQUATION
    [6] BURNUP OR DEPLETION CALCULATION
                        SELECTED NUMBER-->
```

รูป 3.2 ภาพแสดงเมนูหลัก



รูป 3.3 ภาพแสดงเมนูย่อย

คำสั่งภาษาเบสิก ที่สำคัญที่ใช้ในโปรแกรมเชื่อมโยง MAINMENU.BAS และ SUBMENU.BAS

คำสั่ง	คำอธิบาย
COMMON	เป็นคำสั่งส่งค่าตัวแปร จากโปรแกรมหนึ่งไปอีกโปรแกรมหนึ่ง ในภาษาเบสิก
CHAIN "	เป็นคำสั่งปรับการทำงานไปยังโปรแกรมหลังคำสั่ง CHAIN "
SHELL "	เป็นการขอใช้คำสั่งในดอส (disk operating system, DOS) โดยใส่ไว้หลังคำสั่ง SHELL "
A$=INPUT$(1)	รับอินพุทจากคีบอร์ดเป็นข้อมูลสตริงความยาว 1 ตัวอักษร แล้วทำงานต่อทันทีโดยไม่ต้องกด RETURN
SYSTEM	กลับสู่ดอส

โปรแกรมเชื่อมโยงทั้ง 2 โปรแกรม จะใช้คำสั่งเหล่านี้ทำการเชื่อมโยงไฟล์ทั้งหมดเข้าอยู่ในระบบเดียวกัน ทำให้การใช้งานชุดโปรแกรม VPI สะดวกขึ้น

(1) โปรแกรมการอธิบายบทนำ หลังจากเลือกการคำนวณจากเมนูหลักแล้ว ก็จะเข้าสู่เมนูย่อย ซึ่งมีลักษณะเหมือนกันในแต่ละมอดูล ตามภาพแสดงเมนูย่อยข้างต้น ทางเลือกอันแรกในเมนูย่อยคือ บทนำ

(introduction) ซึ่งเป็นบทอธิบายมอดูลที่เลือกจากเมนูหลัก บทอธิบายนี้เขียนโดยใช้ word processor เป็นไฟล์แบบเทกส์ (text file) สามารถขอดูได้โดยใช้คำสั่งในดอสปกติ โปรแกรมการอธิบายบทนำเป็นโปรแกรมภาษาเบสิกชื่อ READTX1.BAS มีขนาดเล็กเขียนขึ้นเพื่อใช้อ่านบทอธิบายแล้วนำมาแสดงผลบนจอภาพ โดยสามารถดูข้อความหน้าถัดไปหรือหน้าก่อน ได้
คำสั่งภาษาเบสิก ที่สำคัญในโปรแกรมนี้

คำสั่ง	คำอธิบาย
ON ERROR GOTO	ถ้ามีความผิดพลาดขึ้น จะกระโดดไปทำงานที่ บรรทัดหลังคำสั่ง GOTO
COMMON	รับค่าตัวแปรมาจากโปรแกรม MAINMENU.BAS
OPEN "AA" FOR INPUT AS #1	เปิดไฟล์ชื่อ AA เพื่อการอ่าน กำหนดเป็นไฟล์ เลขที่ 1
LINE INPUT #1,	อ่านข้อมูลสตริงจากไฟล์เลขที่ 1 ตัวแปรที่ตามหลังคอมมาเป็นตัวแปรสตริง
CLOSE #1	ปิดไฟล์เลขที่ 1
CHAIN "	ส่งการทำงานไปยังไฟล์หลังคำสั่ง CHAIN "

(2) โปรแกรมการเตรียมอินพุท นอกจากการเปลี่ยนรูปแบบในการอ่านข้อมูลเป็นแบบอิสระ (free format) แล้วยังได้เขียนโปรแกรมเพื่อช่วยในการเตรียมข้อมูลอินพุทสำหรับมอดูลในชุดโปรแกรม VP1 แต่ละมอดูล ลักษณะของโปรแกรมเป็นการออกข้อความบนจอภาพ แล้วรอให้ผู้ใช้พิมพ์ค่าที่ต้องการลงไป เช่น

```
Enter fuel rod radius (cm)
.455 (พิมพ์ .455 ตามด้วย RETURN)
```

ในกรณีค่าอินพุทบางตัว ซึ่งผู้ใช้อาจไม่เข้าใจความหมาย โปรแกรมก็จะแสดงข้อความอธิบายความหมายของตัวแปรนั้น ๆ โปรแกรมช่วยการเตรียมข้อมูลมีอยู่ 6 โปรแกรม คือ INPUT1.BAS , INPUT2.BAS , INPUT3.BAS , INPUT4.BAS , INPUT5.BAS และ INPUT6.BAS สำหรับ มอดูล FARCON , SLOCON, DISFAC, ODOG, ODMUG และ FBURN ตามลำดับ โปรแกรมทั้งหมดเขียนด้วยภาษาเบสิก คำสั่งสำคัญที่ใช้ในโปรแกรม ได้แก่

คำสั่ง	คำอธิบาย
ON ERROR GOTO	ถ้ามีความผิดพลาดขึ้น จะกระโดดไปทำงานที่ บรรทัดหลังคำ GOTO

COMMON	เป็นคำสั่งส่งค่าตัวแปรจากโปรแกรมหนึ่งไปอีกโปรแกรมหนึ่งในภาษาเบสิก
INPUT	รับข้อมูลจากคีบอร์ดมาเก็บในตัวแปรที่ตามหลังคำสั่ง INPUT (หลังใส่ข้อมูลแล้วต้องกด RETURN)
GOSUB	ข้ามไปทำโปรแกรมย่อยตามหมายเลขบรรทัดหลัง GOSUB
RETURN	ออกจากโปรแกรมย่อยไปทำต่อในบรรทัดหลังคำสั่ง GOSUB
OPEN "AA" FOR OUTPUT AS #1	เปิดไฟล์ชื่อ AA เพื่อการเขียน กำหนดเป็นไฟล์เลขที่ 1
PRINT #1,	พิมพ์ค่าตัวแปรที่ตามหลังคอมมา ลงในไฟล์เลขที่ 1
OPEN "BB" FOR INPUT AS #2	เปิดไฟล์ชื่อ BB เพื่อการอ่าน กำหนดเป็นไฟล์เลขที่ 2
INPUT #2,	อ่านค่าตัวแปรที่ตามหลังคอมมา จากไฟล์เลขที่ 2
CLOSE #1,#2	ปิดไฟล์เลขที่ 1 และ 2
A$=INPUT$(1)	รับอินพุทจากคีบอร์ด เป็นข้อมูลสตริงความยาว 1 ตัวอักษร แล้วทำงานต่อทันทีโดยไม่ต้องกด RETURN
CHAIN "	เป็นคำสั่งปรับการทำงานไปยังโปรแกรมหลังคำสั่ง CHAIN "

(3) การประมวลผลโปรแกรม เป็นการประมวลผลโปรแกรมตามจุดประสงค์ของการคำนวณที่ได้เลือกไว้ก่อนแล้วในเมนูหลัก โปรแกรมที่จะประมวลผลเป็นมอดูลในชุดโปรแกรม VP1 ซึ่งประกอบด้วย FARCON , SLOCON , DISFAC , ODOG , ODMUG และ FBURN (เปลี่ยนชื่อจากโปรแกรมเดิมคือ FUELBURN) การใช้คำสั่งเพื่อไปประมวลผลโปรแกรมตามที่ต้องการทำได้โดยการกำหนดตัวแปรสตริงขึ้นมาตัวหนึ่งคือ SHEL$ ซึ่งเป็นตัวแปรแทนชื่อโปรแกรมที่จะประมวลผล เช่น เมื่อต้องการประมวลผลมอดูล FARCON ก็ปรับตัวแปร SHEL$ ให้เป็นชื่อ FARCON

```
SHEL$ = "FARCON"
```

แล้วใช้คำสั่ง SHELL ซึ่งเป็นคำสั่งในการติดต่อกับดอส เพื่อให้ทำงานตามคำสั่งที่อยู่หลัง SHELL รูปแบบคำสั่งคือ

```
SHELL SHEL$
```

เมื่อเจอคำสั่งนี้ก็จะไปทำการประมวลผลโปรแกรม FARCON ทันที และถ้าต้องการประมวลผลโปรแกรมอื่นก็เปลี่ยนตัวแปร SHEL$ ใหม่

(4) การแสดงผลลัพธ์ทางจอภาพ ใช้โปรแกรมซึ่งเขียนด้วยภาษาเบสิกในชื่อ READTX2.BAS การทำงานของโปรแกรมคล้ายกับโปรแกรมการอธิบายบทนำ ดังที่ได้กล่าวแล้ว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

(5) การแสดงผลลัพธ์ทางเครื่องพิมพ์ ในการแสดงผลทางเครื่องพิมพ์ได้ใช้คำสั่งของดอสคือ COPY และในการติดต่อกับดอส ใช้คำสั่ง SHELL รูปแบบของคำสั่งก็เช่นเดียวกับตอนประมวลผลคือกำหนดตัวแปรสตริง COP$ เช่นต้องการพิมพ์ผลลัพธ์ของมอดูล FARCON โดยการกำหนดให้ COP$ = "COPY FARCON.OUT LPT1:"
เมื่อต้องการสั่งพิมพ์ ใช้คำสั่ง

```
SHELL COP$
```

เครื่องก็จะส่งผลลัพธ์ของโปรแกรม FARCON ออกทางเครื่องพิมพ์ LPT1:

การปรับปรุงและพัฒนาโปรแกรมดังที่ได้อธิบายไปแล้วนั้น เป็นการปรับปรุงเพื่อให้เกิดความสะดวกในการใช้งานชุดโปรแกรม VPI ในด้านการเตรียมข้อมูล การอธิบายมอดูล การประมวลผล และการแสดงผลลัพธ์ที่ได้จากการประมวลผล ในส่วนของการปรับปรุงมอดูลต่างๆ ในชุดโปรแกรม VPI ก็เพียงเพื่อให้สามารถใช้ได้กับเครื่องไมโครคอมพิวเตอร์ไอบีเอ็มพีซี เท่านั้น โครงสร้างหลักของมอดูลไม่ได้มีการเปลี่ยนแปลง ดังนั้นจึงจัดได้ว่า ชุดโปรแกรมที่ปรับปรุงแล้วยังมีลักษณะเป็นแบบต้นฉบับเดิม (original version) จากโปแลนด์

3.2.5 การเชื่อมโยงมอดูลในชุดโปรแกรม VPI

เนื่องจากลักษณะการทำงานของแต่ละมอดูลในชุดโปรแกรม VPI เป็นแบบอิสระไม่ขึ้นต่อกัน การเตรียมข้อมูลอินพุทต้องกระทำสำหรับแต่ละมอดูล แต่ผลลัพธ์ที่ได้จากมอดูลหนึ่งอาจมีประโยชน์ในการเตรียมอินพุทของมอดูลอื่น โดยจุดประสงค์หลักของผู้พัฒนาโปรแกรมดั้งเดิมนั้นต้องการให้ผู้ใช้มีความเข้าใจในมอดูลแต่ละส่วนอย่างละเอียด แต่ลักษณะดังกล่าวนี้ทำให้เกิดความไม่สะดวกในการใช้งาน เกิดความล่าช้าในการเตรียมข้อมูลสำหรับผู้ใช้ที่เคยศึกษาขั้นต้นมาแล้ว จะเห็นว่าถ้าสามารถที่จะเชื่อมโยงมอดูลต่าง ๆ ในชุดโปรแกรม VPI เข้าด้วยกันได้ก็จะลดความยุ่งยากในการทำงานได้อย่างมาก ซึ่งจะช่วยให้ผู้ใช้สามารถศึกษาปัญหาในด้านการจัดการเชื้อเพลิงด้วยชุดโปรแกรม VPI ได้อย่างมีประสิทธิภาพยิ่งขึ้นหากต้องการใช้อย่างเบ็ดเสร็จ โดยการเตรียมข้อมูลเพียงครั้งเดียว

จากการศึกษาความเป็นไปได้ในการเชื่อมโยง พบว่ามอดูลที่สามารถเชื่อมโยงกันได้ มี 4 มอดูล คือ FARCON , SLOCON , ODMUG และ FBURN โดยลักษณะของการเชื่อมโยงจะเป็นแบบการใช้ข้อมูลอินพุทร่วมกัน การส่งเอาท์พุทจากมอดูลหนึ่งไปเป็นข้อมูลอินพุทของอีกมอดูลหนึ่งอย่างอัตโนมัติ และการรับส่งข้อมูลเพื่อการคำนวณในวัฏจักรเชื้อเพลิงถัดไป

(1) การเตรียมข้อมูลอินพุทสำหรับชุดโปรแกรมรุ่นเชื่อมโยง ข้อมูลอินพุทของแต่ละมอดูลในชุดโปรแกรมรุ่นปกติที่ไม่มีการเชื่อมโยง มีลักษณะเป็นข้อมูลที่จะต้องมีการคำนวณล่วงหน้าก่อนที่จะส่งเป็นข้อมูลอินพุทของแต่ละมอดูล ข้อดีของการเตรียมอินพุทแบบนี้ คือทำให้ผู้ใช้ทราบถึงค่าต่างๆ ที่มอดูลใช้ในการคำนวณ

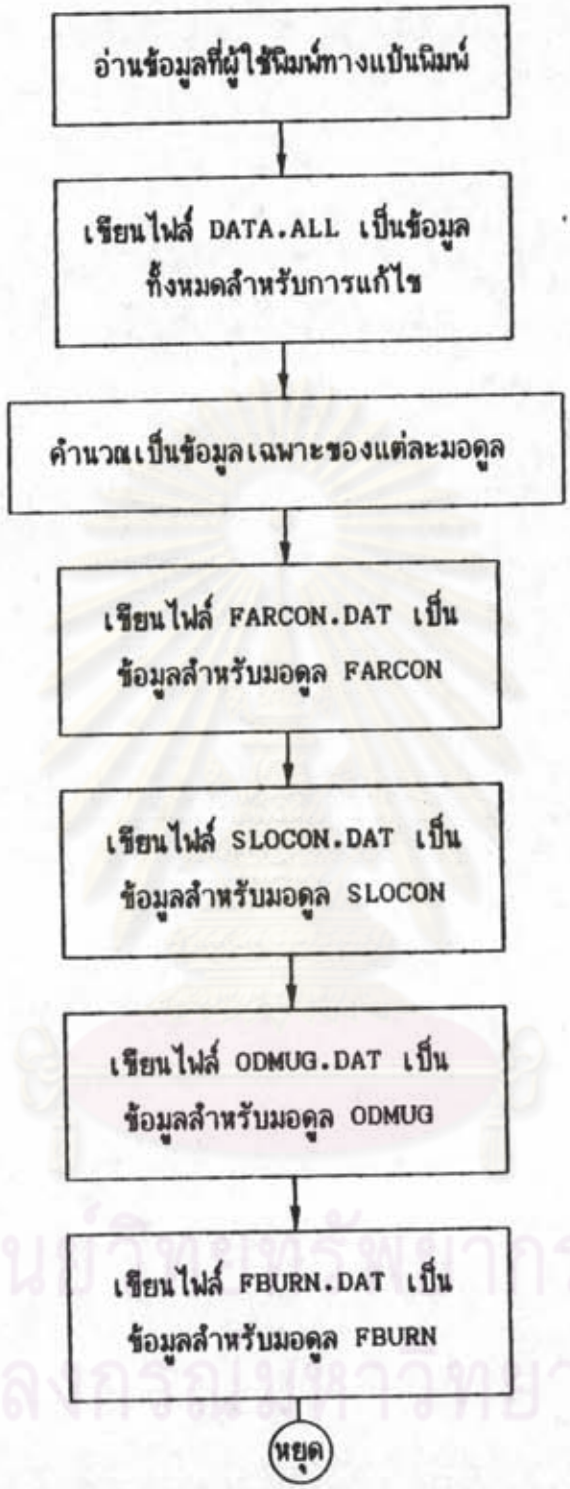
และทำให้ผู้ใช้มีประสบการณ์ในการเตรียมข้อมูล ซึ่งจะมีประโยชน์ในการศึกษาและทำการเตรียมข้อมูลสำหรับโปรแกรมคำนวณเครื่องปฏิกรณ์ที่เป็นมาตรฐานและซับซ้อนยิ่งขึ้นกว่านี้ ส่วนข้อเสียของการเตรียมข้อมูลแบบนี้ก็คือ จำเป็นต้องใช้เวลาในการเตรียมข้อมูลสำหรับปัญหาหนึ่งๆ ค่อนข้างนาน และสำหรับชุดโปรแกรม VPI ยังมีลักษณะการเตรียมข้อมูลที่ซ้ำซ้อน และมีรูปแบบไม่เหมือนกัน ตัวอย่างเช่น ข้อมูลชนิดเดียวกันที่ใช้ในมอดูล FARCON และ SLOCON มีรูปแบบในการเตรียมไม่เหมือนกัน ทำให้เกิดความยุ่งยากกับผู้ใช้

ในชุดโปรแกรมรุ่นเชื่อมโยงได้หลีกเลี่ยงความยุ่งยากในการเตรียม และความซ้ำซ้อนที่เกิดขึ้นดังที่ได้กล่าวแล้ว โดยการที่กำหนดให้มีการเตรียมข้อมูลเพียงครั้งเดียว ลักษณะของข้อมูลจะเป็นลักษณะเฉพาะของแกนกลางเครื่องปฏิกรณ์ เซลล์เครื่องปฏิกรณ์ และวัสดุที่เป็นส่วนประกอบเท่านั้น ข้อมูลเหล่านี้จะถูกคำนวณก่อน จนได้เป็นข้อมูลเฉพาะของแต่ละมอดูลแล้วจึงส่งข้อมูลที่ผ่านการคำนวณไปเก็บยังไฟล์ต่างๆ ในรูปแบบที่มอดูลสามารถอ่านได้

(2) โปรแกรมการเตรียมข้อมูลสำหรับรุ่นเชื่อมโยง เป็นโปรแกรมภาษาเบสิกมีชื่อว่า NEWVER.BAS มีขนาดประมาณ 13300 ไบท์ แผนภาพการทำงานแสดงอยู่ในรูป 3.4 ลักษณะการทำงานคล้ายกับโปรแกรมการเตรียมอินพุทในรุ่นปกติ โดยมีการแสดงข้อความบนจอแล้วรอให้ผู้ใช้นิมน์ค่าที่ต้องการลงไป เช่น

```
core geometry [0]plane [1]cylinder [2]sphere        1
number of fuel zones                                3
external radius (cm) of region # 1                  45
external radius (cm) of region # 2                  90
external radius (cm) of region # 3                  120
thickness (cm) of water reflector( 0 if absent)     0
```

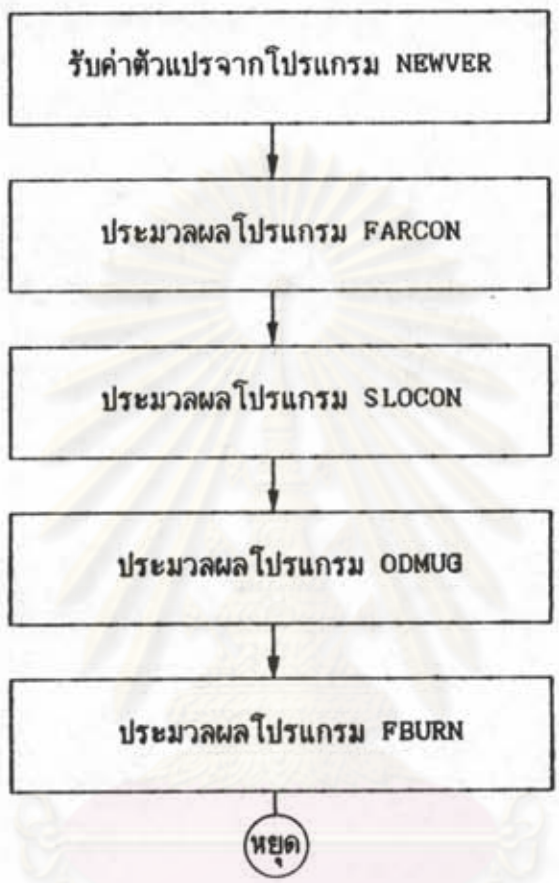
หลังจากใส่ข้อมูลทางแป้นพิมพ์เรียบร้อย โปรแกรมจะทำการคำนวณเพื่อหาค่าข้อมูลเฉพาะของแต่ละมอดูลแล้วจึงเขียนไฟล์ข้อมูลสำหรับมอดูลต่างๆ ไฟล์ข้อมูลของมอดูลในชุดโปรแกรมรุ่นเชื่อมโยงที่เขียนโดยโปรแกรม NEWVER ประกอบด้วย FARCON.DAT เป็นไฟล์ข้อมูลของมอดูล FARCON SLOCON.DAT เป็นไฟล์ข้อมูลของมอดูล SLOCON ODMUG.DAT เป็นไฟล์ข้อมูลของมอดูล ODMUG และ FBURN.DAT เป็นไฟล์ข้อมูลของมอดูล FBURN



รูป 3.4 แผนภาพการทำงานของโปรแกรม NEWVER.BAS

(3) การประมวลผลมอดูลในชุดโปรแกรมรุ่นเชื่อมโยง เมื่อสิ้นสุดการประมวลผล โปรแกรม NEWVER ก็จะส่งค่าตัวแปรและการทำงานให้กับโปรแกรม PLINK โดยโปรแกรม PLINK จะทำหน้าที่ในการสั่งการประมวลผลมอดูลต่างๆ ในชุดโปรแกรมรุ่นเชื่อมโยง ขั้นตอนการประมวลผลมอดูลเรียงตามลำดับคือ

FARCON, SLOCON, ODMUG และ FBURN ลำดับการประมวลผลถูกบังคับให้เป็นแบบนี้เนื่องจากมอดูลที่ประมวลผลหลัง ต้องการข้อมูลซึ่งเป็นผลลัพธ์ ของมอดูลที่ประมวลผลก่อน แผนภาพการทำงานของโปรแกรม PLINK แสดงอยู่ในรูป 3.5



รูป 3.5 แผนภาพการทำงานของโปรแกรม PLINK.BAS

3.2.6 การปรับปรุงมอดูลต่างๆ สำหรับรุ่นเชื่อมโยง

จากการศึกษาการคำนวณการจัดการเชื้อเพลิงโดยใช้ชุดโปรแกรม VPI รุ่นปกติที่ได้ปรับปรุงขึ้นใหม่แสดงให้เห็นว่าสำหรับการเชื่อมโยงมอดูล จำเป็นต้องทำให้แต่ละมอดูลมีความสามารถมากขึ้นดังนี้ สำหรับมอดูล FARCON และ SLOCON ต้องมีความสามารถเพิ่มขึ้น คือ

- สามารถคำนวณได้มากกว่า 1 รีเจียนในการประมวลผลเพียงครั้งเดียว
- สามารถคำนวณภาคตัดขวางมหภาคของการดูดกลืนเฉลี่ยของวัสดุหุ้มแท่งเชื้อเพลิงกับตัวระบายความร้อน (clad - coolant)
- สามารถคำนวณค่าคงที่ของน้ำที่เป็นตัวสะท้อนนิวตรอน (reflector)
- สามารถรับข้อมูลจากไฟล์ที่เขียนโดยมอดูล FBURN

- สามารถ ส่งข้อมูล ให้กับมอดูล FBURN และ ODMUG

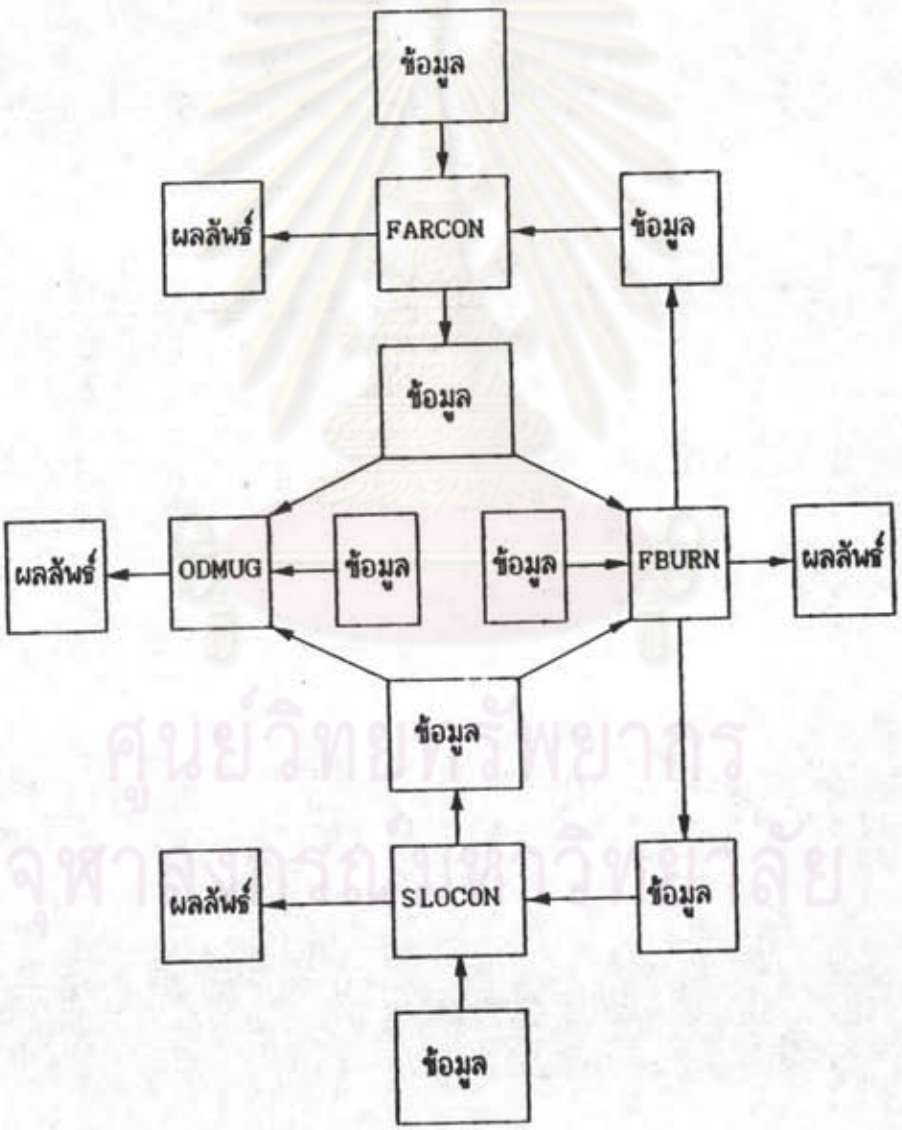
สำหรับมอดูล FBURN ต้องมีความสามารถเพิ่มขึ้น คือ

- สามารถคำนวณได้มากกว่า 1 รีเจียน ในการประมวลผลเพียงครั้งเดียว
- สามารถรับข้อมูลที่เขียนโดยมอดูล FARCON และ SLOCON
- สามารถส่งข้อมูลให้กับมอดูล FARCON และ SLOCON

สำหรับมอดูล ODMUG ต้องมีความสามารถ คือ

- สามารถรับข้อมูลจากมอดูล FARCON และ SLOCON

การเชื่อมโยงการทำงานระหว่างมอดูลได้แสดงเป็นแผนภาพในรูป 3.6



รูป 3.6 แผนภาพแสดงการเชื่อมโยงการทำงานระหว่างมอดูล

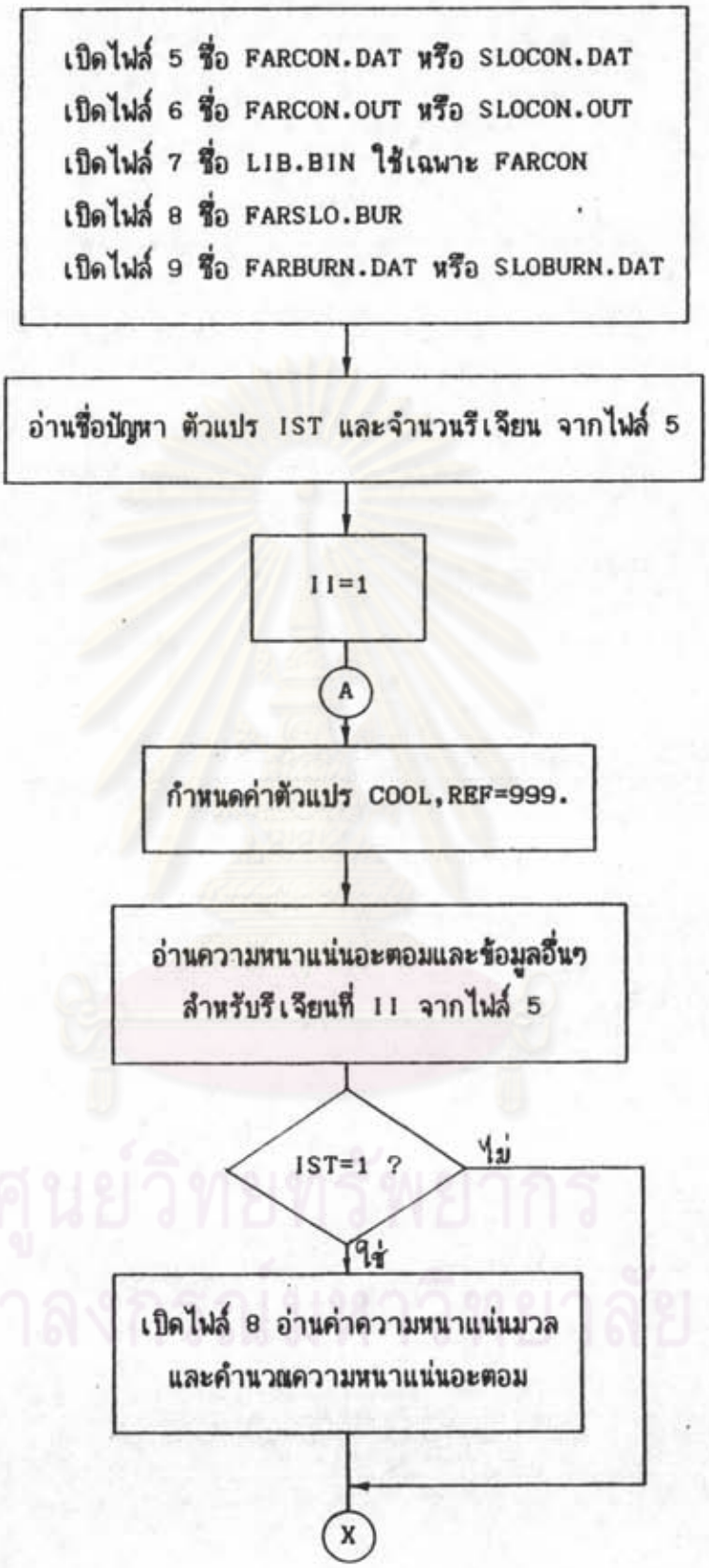
3.2.6.1 การปรับปรุงมอดูล FARCON และ SLOCON

การปรับปรุงให้มอดูล FARCON และ SLOCON สามารถคำนวณได้มากกว่า 1 รีเจียนในการประมวลผลเพียงครั้งเดียวทำได้โดยการบังคับให้มอดูลอ่านข้อมูลของรีเจียน 1 จากไฟล์ข้อมูล แล้วทำการคำนวณจนเสร็จเรียบร้อย แล้วจึงเริ่มอ่านข้อมูลของรีเจียนใหม่ไปทำการคำนวณโดยทำต่อเนื่องกันไปจนครบ สำหรับการคำนวณภาคตัดขวางมหภาคของการดูดกลืนเฉลี่ยของวัสดุหุ้มแท่งกับตัวระบายความร้อนซึ่งเป็นข้อมูลที่มอดูล FBURN ต้องใช้ ทำได้โดยการกำหนดให้เซลล์ประกอบด้วยวัสดุหน่วงนิวตรอนและน้ำเท่านั้น แล้วทำการคำนวณเสมือนเป็นรีเจียนพิเศษ โดยการกำหนดตัวแปรดัชนี COOL ค่าของตัวแปรจะเป็นตัวชี้ว่าเป็นการคำนวณรีเจียนเชื้อเพลิงปกติ หรือรีเจียนพิเศษ

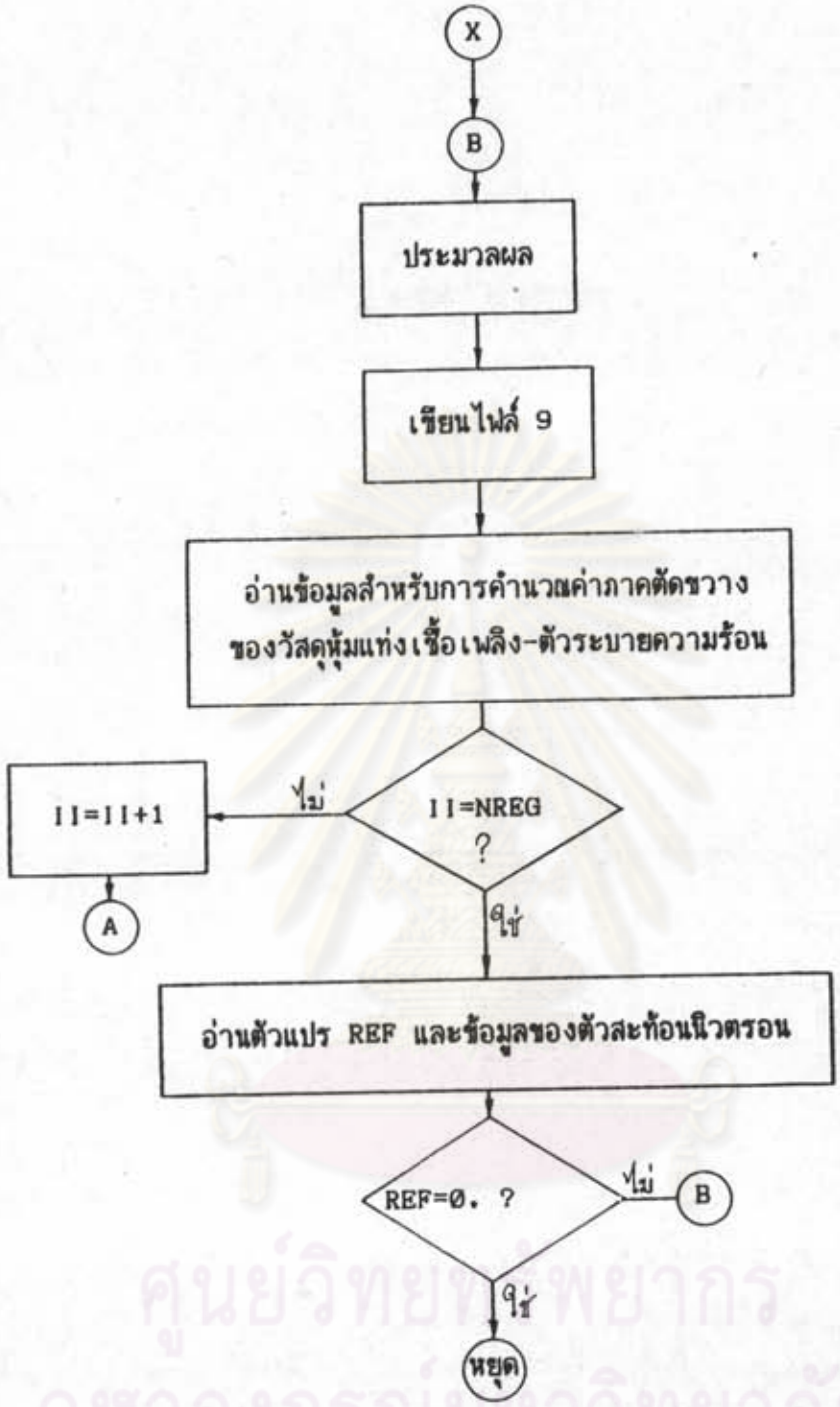
สำหรับในการคำนวณค่าคงที่ของน้ำที่เป็นตัวสะท้อนนิวตรอน ทำได้โดยการกำหนดให้เซลล์ประกอบด้วยน้ำเท่านั้นแล้วทำการคำนวณเสมือนเป็นรีเจียนปกติ โดยการกำหนดตัวแปรดัชนี REF เพื่อชี้ว่า เป็นการคำนวณรีเจียนเชื้อเพลิงหรือตัวสะท้อนนิวตรอน

ในระหว่างการคำนวณของมอดูล FARCON และ SLOCON จะนำเอาผลการคำนวณส่วนหนึ่งไปเขียนเป็นไฟล์ข้อมูลของมอดูล FBURN และ ODMUG โดยเก็บไว้ในชื่อ FARBURN.DAT และ SLOBURN.DAT ตามลำดับ โครงสร้างใหม่ของมอดูล FARCON และ SLOCON รุ่นเชื่อมโยง ได้แสดงอยู่ในรูป 3.7

การทำงานเริ่มจากการเปิดไฟล์ข้อมูลจากแหล่งต่างๆ เช่น ไฟล์ข้อมูลหลัก (FARCON.DAT สำหรับ FARCON และ SLOCON.DAT สำหรับ SLOCON) ไฟล์ไลบรารี LIB.BIN สำหรับ FARCON ไฟล์เอาท์พุท (FARCON.OUT สำหรับ FARCON และ SLOCON.OUT สำหรับ SLOCON) ไฟล์เอาท์พุทที่จะส่งเป็นอินพุทให้มอดูล FBURN (FARBURN.DAT สำหรับ FARCON และ SLOBURN.DAT สำหรับ SLOCON) และในกรณีที่เป็นการคำนวณในวัฏจักรถัดไปก็จะต้องอ่านข้อมูลที่เป็นเอาท์พุทของมอดูล FBURN คือไฟล์ FARSLO.BUR ซึ่งใช้ร่วมกันทั้งมอดูล SLOCON และ FARCON ต่อไปเริ่มทำการอ่านข้อมูลค่าคงที่จากไฟล์ข้อมูลหลัก แล้วกำหนดค่าตัวแปร COOL และ REF เท่ากับ 999. เพื่อเป็นดัชนีแสดงว่าเป็นรีเจียนเชื้อเพลิง อ่านข้อมูลของรีเจียน II แล้วทำการคำนวณ เมื่อเสร็จก็เขียนไฟล์ FARBURN.DAT (หรือ SLOBURN.DAT สำหรับ SLOCON) ต่อไปทำการอ่านค่า COOL ของรีเจียน II แล้วกลับขึ้นไปคำนวณค่าภาคตัดขวางของการดูดกลืนเฉลี่ยวัสดุหุ้มแท่งเชื้อเพลิงกับตัวระบายความร้อน การคำนวณจะทำซ้ำจนครบทุกรีเจียนเชื้อเพลิง หลังจากนั้นจะอ่านค่าตัวแปรต่างๆ ของตัวสะท้อนนิวตรอน เช่น REF แล้วทำการคำนวณค่าคงที่ต่างๆ ของตัวสะท้อนนิวตรอน เป็นอันเสร็จสิ้นการทำงานทั้งหมด



รูป 3.7 โครงสร้างใหม่ของมอดูล FARCON และ SLOCON สำหรับรุ่นเชื่อมโยง

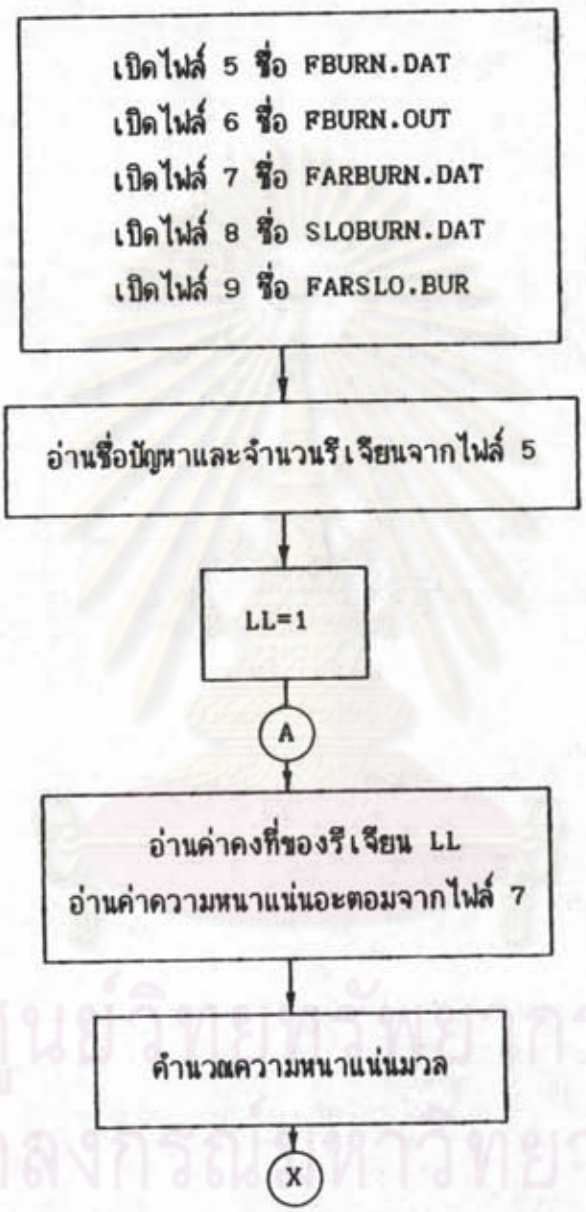


รูป 3.7 โครงสร้างใหม่ของมอดูล FARCON และ SLOCON สำหรับรุ่นเชื่อมโยง (ต่อ)

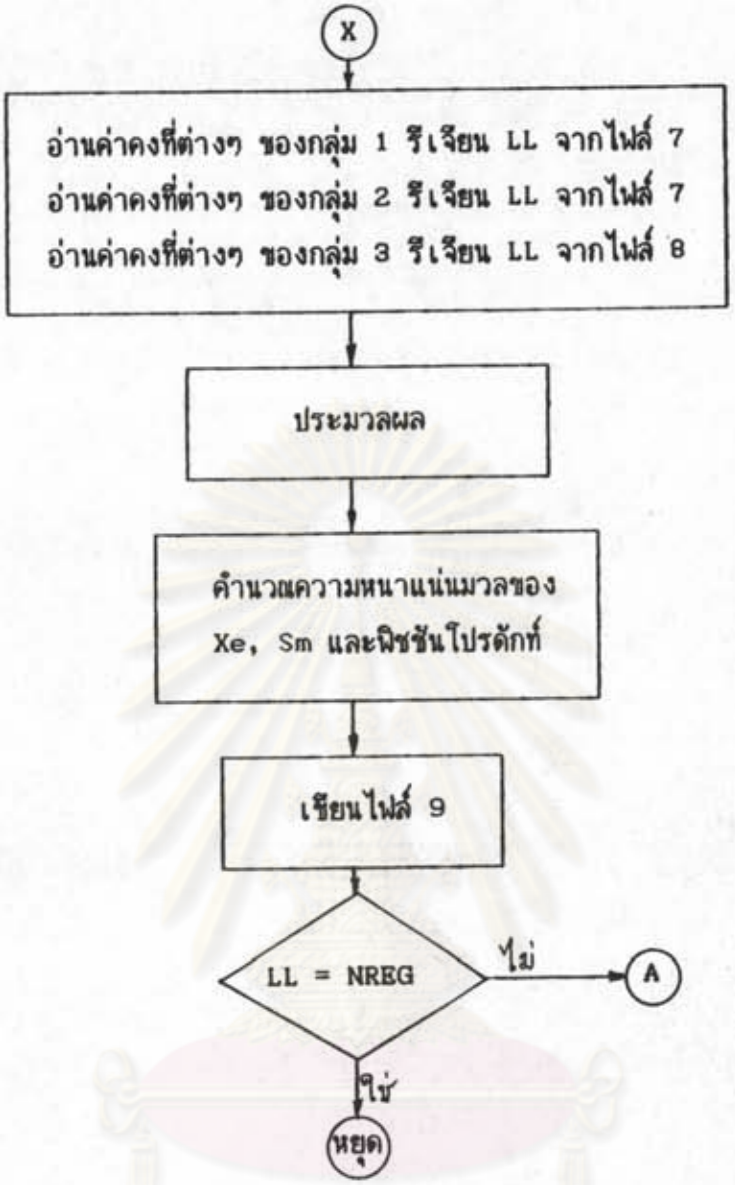
3.2.6.2 การปรับปรุงมอดูล FBURN สำหรับรุ่นเชื่อมโยง

มอดูล FBURN รุ่นเชื่อมโยง จะต้องมีความสามารถคำนวณได้มากกว่า 1 รีเจียน สามารถรับข้อมูลที่เขียนด้วยมอดูล FARCON และ SLOCON ได้และสามารถเขียนไฟล์ข้อมูลสำหรับวัฏจักรใหม่ให้กับมอดูล FARCON และ SLOCON ได้ ไฟล์ที่เปิดใช้ในมอดูล FBURN ประกอบด้วย ไฟล์ FBURN.DAT เป็นไฟล์ข้อมูลหลัก ไฟล์ FBURN.OUT เป็นไฟล์เก็บผลลัพธ์ ไฟล์ FARBURN.DAT และ SLOBURN.DAT เป็น

ไฟล์ที่เขียนโดยมอดูล FARCON และ SLOCON ตามลำดับ ไฟล์ FARSLO.BUR เป็นไฟล์เอาท์พุทที่เขียนให้กับมอดูล FARCON และ SLOCON เพื่อเป็นข้อมูลในการคำนวณสำหรับวัฏจักรถัดไป โครงสร้างใหม่ของมอดูล FBURN แสดงในรูป 3.8



รูป 3.8 โครงสร้างใหม่ของมอดูล FBURN สำหรับรุ่นเชื่อมโยง



รูป 3.8 โครงสร้างใหม่ของมอดูล FBURN สำหรับรุ่นเชื่อมโยง (ต่อ)

ปัญหาของการเชื่อมโยงมอดูล FBURN เพื่อที่จะเพิ่มประสิทธิภาพในการทำงานของชุดโปรแกรม VPI ให้มากยิ่งขึ้น ด้วยการจัดการให้มอดูล FBURN ส่งเอาท์พุทของตัวเองไปเป็นอินพุทของมอดูล FARCON และ SLOCON โดยเก็บไว้ในไฟล์ชื่อ FARSLO.BUR เพื่อให้มอดูล FARCON และ SLOCON ใช้เป็นข้อมูลในการคำนวณวัฏจักรเชื้อเพลิงอันใหม่ เอาท์พุทของมอดูล FBURN ที่จะส่งเป็นอินพุทของมอดูล FARCON และ SLOCON จะต้องประกอบด้วย ความหนาแน่นมวล (mass density) ของไอโซโทปฟิสไซล์ (fissile) และเฟอร์ไทล์ (fertile) ที่เปลี่ยนไปหลังจากเบิร์นอัพ และความหนาแน่นมวลของฟิชชันโปรดักท์ที่สำคัญ คือ ซาแมเรียม-149 และ ซีนอน-135 ซึ่งเป็นข้อมูลที่มอดูล FARCON และ SLOCON ใช้ในการคำนวณ

ปัญหาที่พบในขั้นตอนนี้คือ การที่ไม่ทราบค่าภาคตัดขวางจุลภาคของการดูดกลืนในช่วงเทอร์มัลของไอโซโทปซาแมเรียม-149 ทราบแต่เพียงค่าของซีนอน-135 เท่านั้น จึงต้องทำการคำนวณหาค่าภาคตัดขวาง

จุลภาคของการดูดกลืนในช่วงเทอร์มัลของ ซาแมเรียม-149 ในการคำนวณพิจารณาให้นิวตรอนและตัวดูดกลืนมีการแจกแจงแบบแมกซ์เวลล์ (Maxwellian distribution) ที่ช่วงพลังงานเทอร์มัล คำนวณโดยใช้แฟกเตอร์แก้แบบ non-1/V (non-1/V factor, G(T)) ทำการหาว่าที่อุณหภูมิเท่าไรจึงทำให้ซีนอน-135 มีภาคตัดขวางจุลภาคเฉลี่ยของการดูดกลืนเท่ากับ 1.53×10^{6} บาร์น ซึ่งเป็นค่าที่ใช้ในมอดูล FBURN ภาคตัดขวางจุลภาคของการดูดกลืนของซีนอน-135 ที่นิวตรอนพลังงาน .0253 อิเล็กตรอนโวลต์ เท่ากับ 2.7×10^{6} บาร์น (4)

$$1.53\times10^{6} = \frac{G(T)}{1.128} \times 2.7\times10^{6} \sqrt{\frac{.0253}{8.617\times10^{-5}T}}$$

ในการคำนวณค่า G(T) ใช้การหาค่าแทรกเชิงเส้น (linear interpolation) จากตาราง 3.1 แสดงค่า G(T) เทียบกับอุณหภูมิ T พบว่าที่อุณหภูมิประมาณ 590 เซลเซียส (celcius) ซีนอน-135 จะมีภาคตัดขวางจุลภาคของการดูดกลืนเท่ากับ 1.530×10^{6} บาร์น

ตาราง 3.1 แฟกเตอร์ non-1/V ของซีนอน-135 และ ซาแมเรียม-149 (1)

T, °C	Xe-135	Sm-149
400	1.1864	2.1854
600	1.0914	2.0854
800	0.9887	1.9246

จากอุณหภูมินี้ทำการคำนวณหาค่า G(T) ของซาแมเรียม-149 ได้เท่ากับ 2.0904 คำนวณค่าภาคตัดขวางที่ 590 เซลเซียส (จากค่าภาคตัดขวางที่ พลังงาน .0253 อิเล็กตรอนโวลต์ เท่ากับ 4.18×10^{4} บาร์น) จากสูตร (1)

$$\sigma_{sm}(863\ K) = \frac{2.0904}{1.128} \times 4.18\times10^{4} \sqrt{\frac{.0253}{8.617\times10^{-5}\times863}}$$

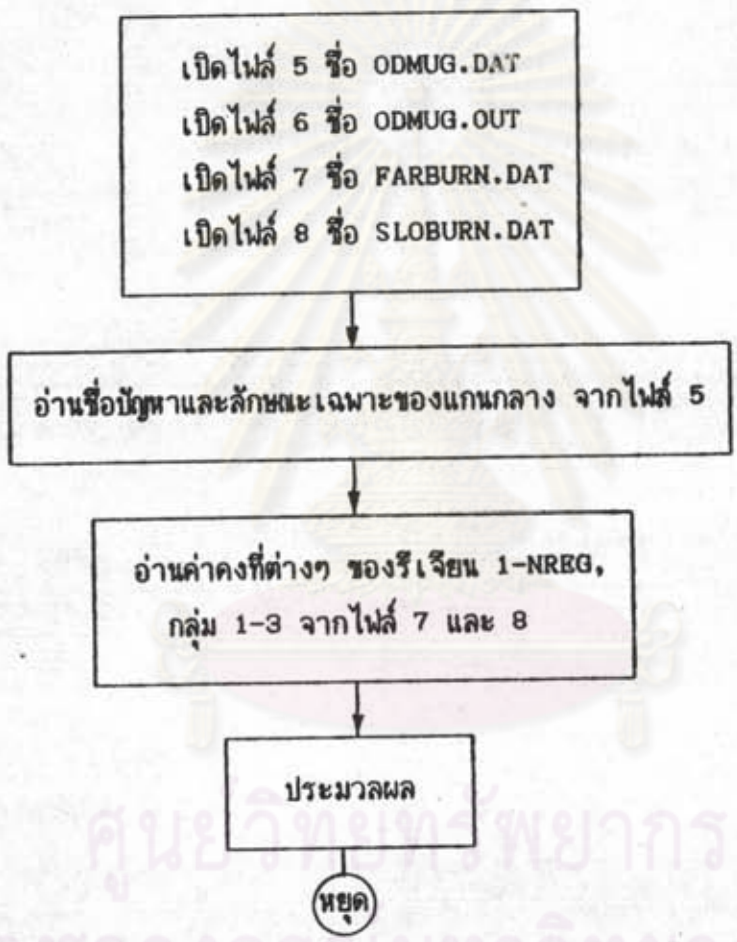
$$= 4.52\times10^{4} \quad \text{บาร์น}$$

สำหรับในส่วนของฟิชชันโปรดักท์ เนื่องจากในมอดูล FBURN ได้แยกพิจารณาฟิชชันโปรดักท์ออกเป็น 2 กลุ่ม คือ กลุ่มที่เกิดการอิ่มตัวอย่างช้าๆ (slowly saturating) และกลุ่มที่ไม่เกิดการอิ่มตัว (non-saturating) แต่ในมอดูล SLOCON ไม่ได้แบ่งแยกไว้โดยคิดรวมเป็นกลุ่มเดียว ในการคำนวณจึงพิจารณาโดยประมาณให้ฟิชชันโปรดักท์รวมเท่ากับผลบวกของทั้งสองกลุ่มและส่งผลรวมนี้ไปให้มอดูล SLOCON

ทำการคำนวณสำหรับวัฏจักรถัดไป

3.2.6.3 การปรับปรุงมอดูล ODMUG สำหรับรุ่นเชื่อมโยง

มอดูล ODMUG ในรุ่นเชื่อมโยงต้องสามารถรับข้อมูลที่เป็นเอาท์พุทของมอดูล FARCON และ SLOCON มาทำการประมวลผลได้ ไฟล์ข้อมูลของมอดูล ODMUG ประกอบด้วย ไฟล์ ODMUG.DAT ซึ่งเป็นไฟล์ข้อมูลหลัก ไฟล์ ODMUG.OUT เป็นไฟล์สำหรับเก็บผลลัพธ์ ไฟล์ FARBURN.DAT และ SLOBURN.DAT เป็นไฟล์ข้อมูลที่เขียนโดยมอดูล FARCON และ SLOCON ตามลำดับ เพื่อให้กับมอดูล FBURN และ ODMUG ใช้ร่วมกัน โครงสร้างใหม่ของมอดูล ODMUG แสดงในรูป 3.9



รูป 3.9 โครงสร้างใหม่ของมอดูล ODMUG สำหรับรุ่นเชื่อมโยง