

บทที่ 1

วงการกำงำนของ VAX RPG II

RPG II บนเครื่อง VAX นั้น มีลักษณะเช่นเดียวกับ RPG II บนเครื่อง IBM ดังที่นักศึกษาเคยทราบมาแล้วว่า คุณลักษณะที่ดีของภาษา RPG ก็คือ เป็นภาษาที่สร้างรูปแบบรายงานได้มากมายรูปแบบ ในขณะที่เดียวกันที่สามารถที่จะนำไปประมวลผลกับงานทางธุรกิจได้อย่างมีประสิทธิภาพ

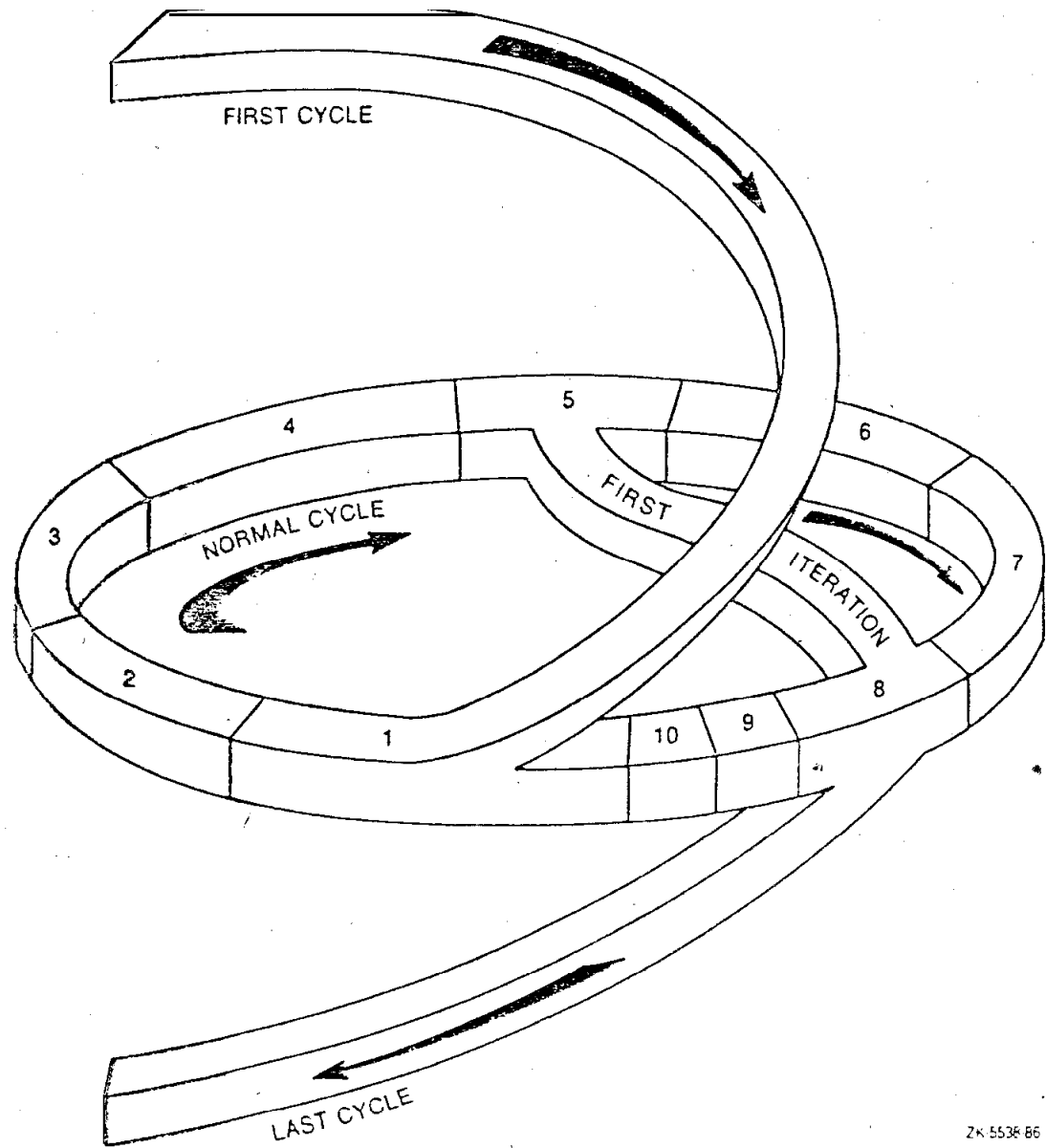
ในกรณีของ RPG II บนเครื่อง VAX นั้น บริษัท DIGITAL ได้พัฒนาและปรับขยาย RPG II ให้ผนวกเข้ากับสถาปัตยกรรมของ VAX/VMS โดยที่ระบบ VAX RPG II จะประกอบด้วยตัวคอมไพเลอร์, text editor พิเศษ, run time support และ conversion utility

ภาษา RPG II นั้น ถือได้ว่าเป็นภาษาประเภท nonprocedural ซึ่งหมายความว่า โปรแกรมที่เขียนภายใต้ภาษา RPG นั้นจะปฏิบัติงานภายใต้กฎเกณฑ์ที่มีแบบแผนคงที่ คือ logic การทำงานจะถูกสร้างให้อยู่ในคอมไพเลอร์ของ RPG อยู่แล้วโดยที่โปรแกรมเมอร์ไม่ต้องคิดสร้าง logic เอง logic cycle ของ RPG นั้นจะอยู่ในรูปที่เรียกว่า built-in logic

ดังนั้น ผู้เขียนโปรแกรมภาษา RPG จึงเพียงแต่เขียนโปรแกรมภายใต้แบบฟอร์มของภาษา RPG เท่านั้น ภายใต้แบบฟอร์มหรือที่เรียกว่า specification ต่าง ๆ นั้น ผู้เขียนโปรแกรมก็เพียงแต่เขียนคำสั่งอธิบายถึงลักษณะของข้อมูลเข้า ข้อมูลออก ปฏิบัติการทางคณิตศาสตร์ และปฏิบัติการพิเศษต่าง ๆ เพื่อดำเนินการกับข้อมูลเข้าจนกว่าจะได้รับข้อมูลสำเร็จรูปแสดงผลตามที่ต้องการ specification ของภาษา RPG จะประกอบด้วยทั้งหมด 7 แบบด้วยกัน ซึ่งรายละเอียดของแต่ละแบบฟอร์มนั้นได้อธิบายอยู่แล้วในหนังสือภาษา RPG ภาคทฤษฎี

การทำงานภายใต้ logic ของ RPG นั้นจะเรียงตามลำดับเป็นแบบแผนตามลักษณะของงานทางด้านธุรกิจ การทำงานต่าง ๆ จะอยู่ภายใต้การควบคุมของ switching ซึ่งเรียกว่า indicator

Figure 1-1: Logical Flow of the VAX RPG II Logic Cycle



Key to Figure 1 -1:

In the FIRST CYCLE, several initialization tasks are accomplished. Files are opened, the local data area is loaded, the system date is obtained from the system, and counters, tables, and arrays are initialized. The program then transits into the NORMAL CYCLE.

- ❶ When the NORMAL CYCLE is entered, program instruction lines with special conditions for Output specifications are processed.
- ❷ All control-level and record-identifying indicators are set off.
- ❸ A record is read, and the appropriate record-identifying indicator is set on.
- ❹ If the control field of the record just read is different from the control field of the previous record, a control *break* occurs. When a control break does occur, the appropriate control-level indicator is set on, plus all lower control-level indicators.
- ❺ If VAX RPG II detects that this is the first iteration after the FIRST CYCLE, it branches to step 8.
- ❻ Total calculations (conditioned by control-level indicators in the Calculation specifications) are processed if the appropriate control-level indicators are on.
- ❼ If totals are requested in the program, the program does total output operations.
- ❽ If the last-record indicator is on, the LAST CYCLE is entered and the program ends.
- ❾ Detail calculations and program output are processed from data read in this cycle.
- ❿ All remaining detail calculations are processed on the data from the current record read at the beginning of the NORMAL CYCLE. Then, the cycle continues around, repeating while there are records to process.

Upon detecting the last-record indicator, the program enters the LAST CYCLE, does one-time cleanup work including total calculations and total output operations, and the program ends.

เราจะแบ่งวงจรการทำงานของภาษา RPG ออกมาเป็น 2 Cycle คือ First Cycle และ Normal Cycle โดยที่ใน First Cycle จะประกอบด้วยงานหลายประเภทด้วยกัน คือ มีการเปิดแฟ้มข้อมูล local data area จะถูก load เข้ามา system date จะถูกรับโดย system พร้อมกันนั้นก็จะมี การสร้างเคาน์เตอร์, ตาราง (table) และอะเรย์ (array) ถัดจากนั้นไปก็จะถึงขั้นตอนของ Normal Cycle เราอาจจะสรุปรายละเอียดวงจรการทำงานของภาษา RPG ดังนี้ คือ

1. เมื่อเข้ามาในช่วงของ Normal Cycle นั้น คำสั่งต่าง ๆ ในโปรแกรมซึ่งก่อให้เกิดเงื่อนไขพิเศษสำหรับผลที่จะออกใน Output specification form จะถูกดำเนินการ

2. ข้อมูลจะถูกรับเข้ามา โดยที่จะมีการตรวจสอบเพื่อเปิด indicator บางตัวอันสืบเนื่องมาจากเงื่อนไขของ record กรณีนี้คือ record indentifying indicator
3. ในกรณีที่มีการกำหนดเงื่อนไขของ level break ใน control field แล้ว ก็จะมีการตรวจสอบว่า field ดังกล่าวของ record ที่เพิ่งเข้ามาใหม่นั้นแตกต่างจาก control field ของ record ก่อนหน้านั้นหรือไม่
4. ถ้าหากว่ามีความแตกต่างกัน level break จะเกิดขึ้น โดยที่ level break ที่สูงกว่าจะมีผลกระทบให้ level break ที่ต่ำกว่า on ด้วย
5. ในกรณีที่วงจรการทำงานของ RPG นั้นกระทำกับ record แรกในแฟ้มข้อมูล ซึ่งเราเรียกว่า First Cycle แล้ว เครื่องจะกระโดดข้ามไปดำเนินงานในขั้นที่ 8
6. การทำงานในช่วงที่เรียกว่า Total Calculation หรือ Total Time นั้น จะหมายถึงการทำงานอันเนื่องมาจากเงื่อนไขของ level indicator ซึ่งอาจจะโยงไปเพื่อจุดมุ่งหมายในการคำนวณหรือการแสดงผล output ในขั้นที่ 7
7. ถ้าผลของ level indicator ต้องการให้ออกผลทาง output เครื่องจะดำเนินงานให้ผลซึ่งเรียกว่า total output
8. ถ้าหากว่า last-record indicator ได้เกิดขึ้น นั้นหมายความว่าวงจรการทำงานของ RPG ได้ถึง Last Cycle แล้ว ดังนั้น ถ้ามีการควบคุมการทำงานอะไรที่เกี่ยวข้องกับ LR แล้ว จะเกิดขึ้น หลังจากนั้นโปรแกรมก็จะจบการทำงาน
9. ช่วงที่เรียกว่า detail time หรือ detail cycle จะหมายถึงการที่ข้อมูลจาก record ได้ถูกส่งเก็บไปยัง processing area และถูกนำไปคำนวณหรือประมวลผล แล้วส่งผลออกทาง output
10. ช่วงที่เริ่มตั้งแต่ข้อมูลถูกรับเข้ามาใน processing area และประมวลผลและออก output จนกระทั่งจบการทำงานของข้อมูลใน record นั้น แล้วเตรียมที่จะรับข้อมูล record ใหม่เข้ามาเรียกว่า Normal Cycle

ในแต่ละ Cycle ของ RPG เราจะสรุปปฏิบัติการต่าง ๆ ได้ดังนี้

First Cycle เริ่มตั้งแต่ปฏิบัติการต่าง ๆ ก่อนหน้าที่ record แรกในแฟ้มข้อมูลจะเข้ามาใน Normal Cycle ปฏิบัติการต่าง ๆ ดังกล่าวจะประกอบด้วย

● รับระบบ ๑ ต่าง ๑ ซึ่งเกี่ยวกับการทำงานซึ่งเรียกว่า Current system เช่น VDATE, VDAY, VMONTH และ VYEAR

- load the local data area
- เปิดแฟ้มข้อมูลทุกแฟ้ม
- รับตารางข้อมูล (table) และอะเรย์
- สร้าง page number ในเคาน์เตอร์
- พิมพ์ heading line และ detail line โดยเงื่อนไขของ IP หรือโดยเงื่อนไขของ negative indicator หรือโดยเงื่อนไขของการไม่มี indicator ใด ๆ ควบคุม

โดยสรุปแล้ววงจรการทำงานของ RPG (logic cycle) นั้นจะประกอบด้วยขั้นตอนของ total time (ยกเว้นเฉพาะกรณีของ First Record และ Last Record) และ Detail Time

Last Cycle

จะเกิดขึ้นภายหลังจากที่ข้อมูลจาก primary file และ secondary file ได้ถูกรับไปปฏิบัติการหมดแล้ว และเมื่อเงื่อนไขของ LR ได้ถูก set on แล้ว จะยังผลให้บรรดา level indicator ทั้งหมด คือ L1-L9 (ถ้ามี) ถูก set on ตามไปด้วย ภายในช่วงของ Last Cycle นั้นจะมีปฏิบัติการดังนี้

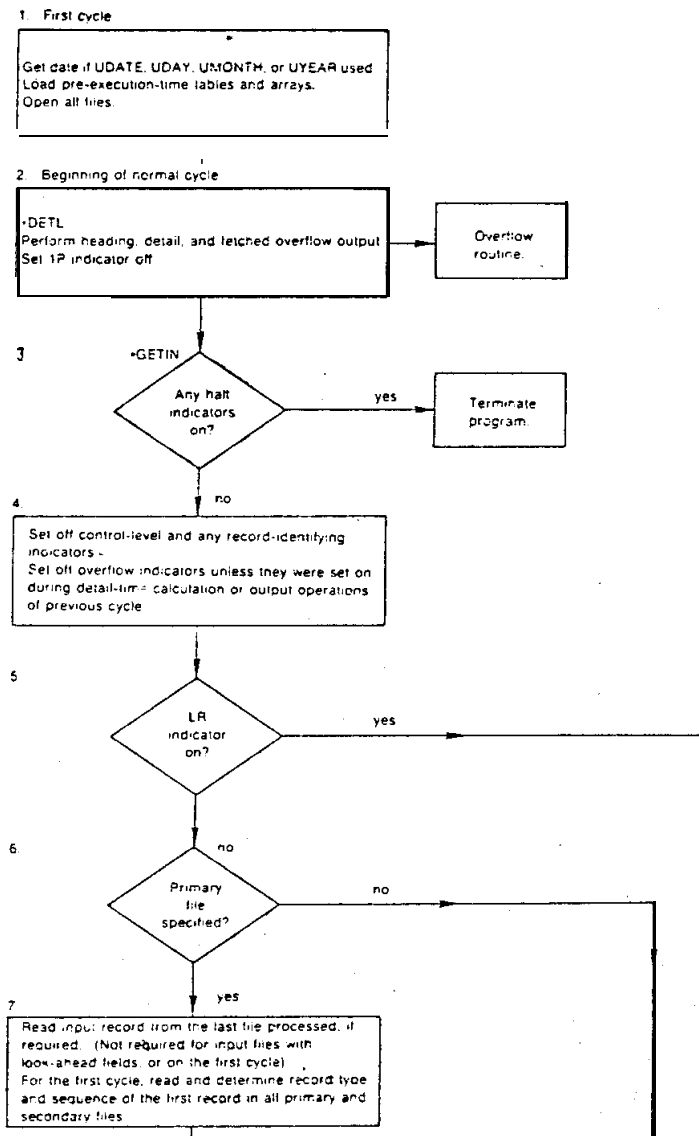
- ปฏิบัติการ total-time calculations
- ปฏิบัติการของการพิมพ์ total-time output
- ให้ผล output ของ arrays หรือ tables ถ้ามีผลสืบเนื่องถึง output file
- ปิดไฟล์ทุกไฟล์
- ให้ผลอันสืบเนื่องมาจาก local data area
- จบปฏิบัติการ

Normal Cycle

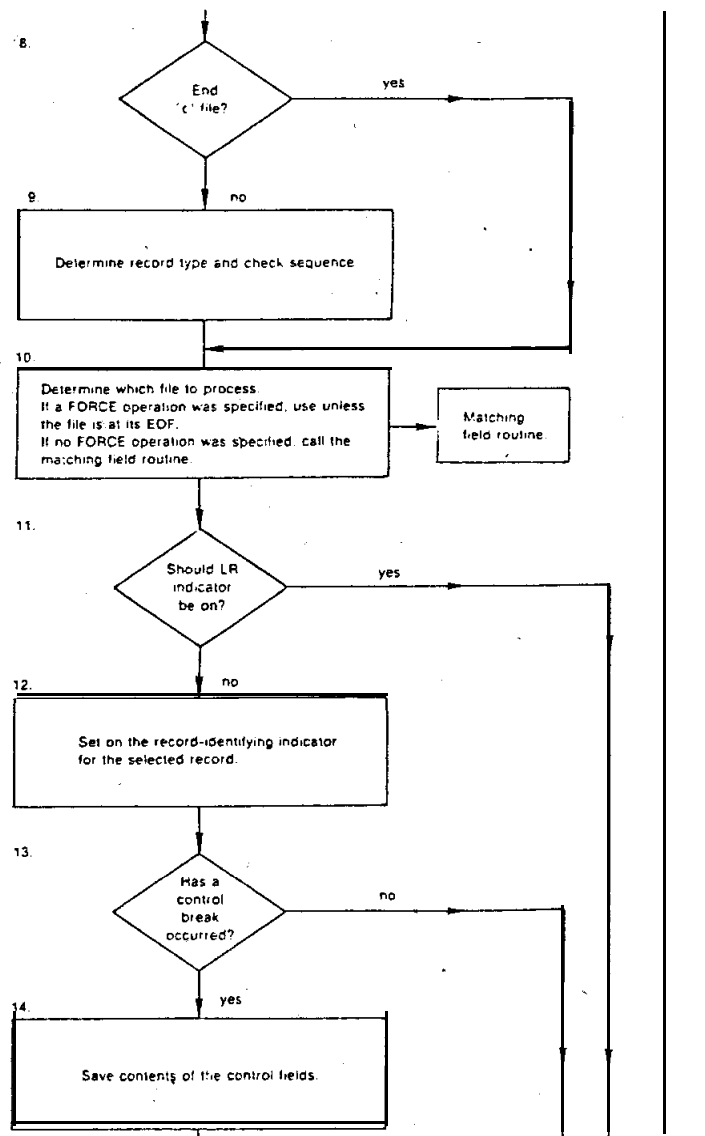
Normal Cycle คือ Cycle ของ RPG ซึ่งไม่ใช่กรณีของ First Cycle หรือ Last Cycle เราจะสรุปการปฏิบัติงานในช่วงของ Normal Cycle ได้ดังนี้

1. ให้ output heading line ถ้ามี
2. ให้ output detail-time อันเป็นผลสืบเนื่องมาจาก record ก่อนหน้านั้น
3. รับ record จากแฟ้มข้อมูลเข้ามาปฏิบัติการ

Figure I-2: Detailed VAX RPG II Logic Cycle Flowchart

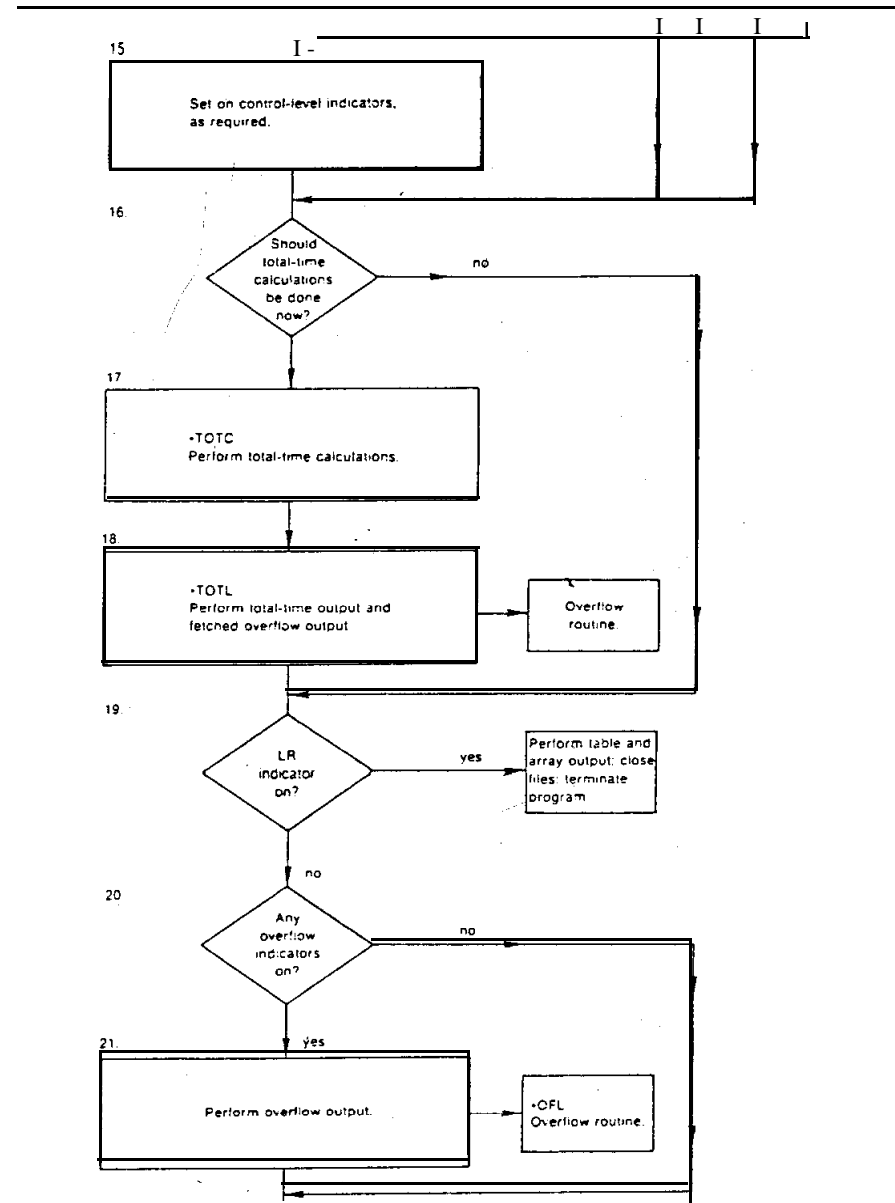


Figure' I-2 (Cont.): Detailed VAX RPG II Logic Cycle Flowchart



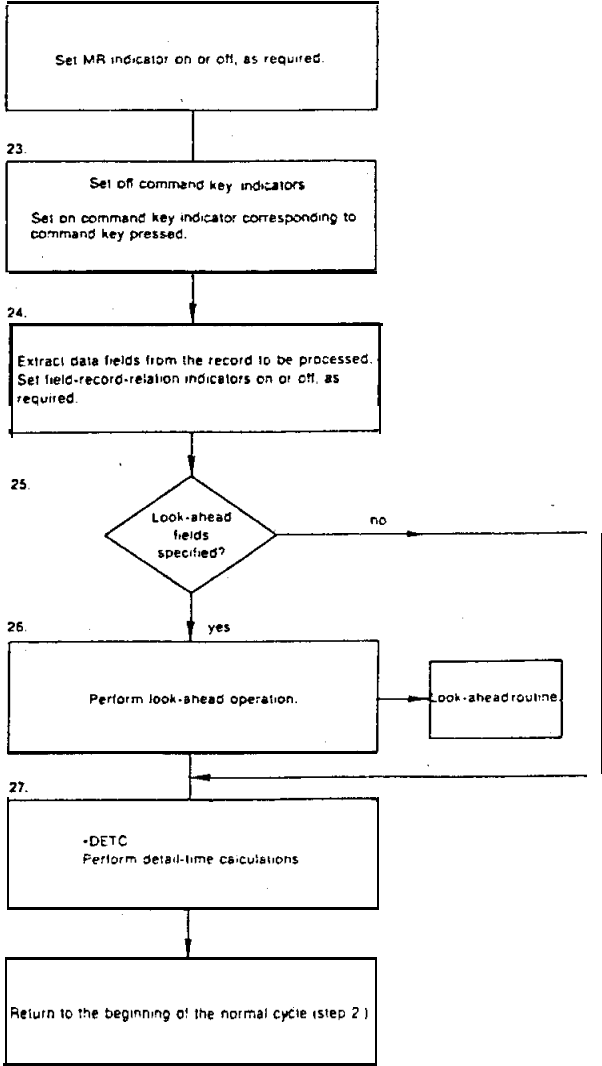
(Continued on next page)

Figure I-2 (Cont.): Detailed VAX RPG II Logic Cycle Flowchart



(Continued on next page)

Figure 1-2 (Cont.): Detailed VAX RPG II Logic Cycle Flowchart



2A-1571-84

Key to Figure 1-2:

- ❶ This step is executed only during the first cycle. It initializes **your** program for execution. Initialization consists of retrieving the date (if you specified UDATE, UDAY, UMONTH, or UYEAR), opening all files, and loading all preexecution-time tables and arrays.
- ❷ VAX RPG II writes heading and detail lines (identified by H or D in column 15 (Type) of the Output specification). Heading and detail lines are always executed at the same time. If conditioning indicators are specified, the conditions for the indicator must be satisfied. If the fetch overflow logic is specified and the overflow indicator is on, RPG II writes the appropriate overflow lines. If the 1P indicator is on (during the first cycle only), VAX RPG II prints all lines conditioned by it, then sets the 1P indicator off. VAX RPG II executes this step at the beginning of the program so that heading lines can be printed before processing begins.
- ❸ VAX RPG II checks whether any halt indicators (HI through H9) are on; if any are, the program terminates. If you do not want your program to terminate here, you must set all halt indicators off previous to this step. You can set halt indicators on, however, at any time during the program.
- ❹ VAX RPG II sets control-level indicators (L1 through L9) and all indicators used as record-identifying indicators off. VAX RPG II also sets overflow indicators (OA through OG, and OV) off, unless they were set on during detail time (detail-time calculations or output operations) in the preceding cycle. All other types of indicators that are on remain on.
- ❺ Here, VAX RPG II determines whether the LR indicator is on. If it is, RPG II branches to step 15 and sets on control-level indicators L1 through L9, if used.
- ❻ VAX RPG II determines whether a primary file was specified by the program. If not, VAX RPG II proceeds directly to step 16.
- ❼ If required, VAX RPG II reads an input record from the last primary or secondary file processed. If this was an input file with look-ahead fields, the record is already available; therefore, no read operation may be necessary at this time. On the first cycle, a record is read from each primary and secondary file.
- ❽ VAX RPG II tests the file just read for end-of-file. If end-of-file has been encountered, the program bypasses step 9.
- ❾ If VAX RPG II reads a record from a file, the record type is determined and the record sequence is checked. If the record type cannot be determined, or the record is out of sequence, the program terminates.
- ❿ In this step, VAX RPG II determines which file to process. If a FORCE operation was executed during the previous cycle, the forced file is selected for processing. (All records processed with a FORCE operation are processed with the matching-records (MR) indicator set off.) However, if the forced file is at end-of-file (EOF), the normal multfile logic selects the next record for processing. If no forced file was specified, VAX RPG II determines whether matching fields were specified. If so, the matching-fields routine is given control (see

Figure 1-3). Otherwise, all records in a primary file are processed first, then the records from each secondary file in order of their specification.

- ⑪ Here, VAX RPG II determines whether the LR indicator should be set on. The LR indicator is set on when the program has reached the end of all the files that you have specified for processing **until** the **end-of-file**, and **when** all **the** records from secondary files that match the last **primary** record have been processed. If the LR indicator should be set on, VAX RPG II branches to step 15 and sets on indicators L1 through L9.
- ⑫ VAX RPG II sets on **the** record-identifying indicator for the **record** selected for processing.
- ⑬ VAX RPG II determines whether the record selected for processing has caused a control break to occur. A control break occurs when the value in the control field of **the** record being processed differs from the previous value of the control field. See Section 1.4.1 for more information.
- ⑭ If a control break has occurred, VAX RPG II saves the contents of all appropriate control fields.
- ⑮ If a control break has occurred, VAX RPG II sets the appropriate control-level indicator (L1 through L9) on; at the same time, VAX RPG II sets all lower-level control-level indicators on. The L1 through L9 indicators can be used for conditioning only if they have been defined as **conditioning** indicators.
- ⑯ VAX RPG II determines whether total-time calculation and output operations should be performed. If control-level indicators are not specified in columns 59 and 60 (control-level) of the Input specification, VAX RPG II bypasses total-time calculation and output operations during the first cycle only; after **the** first cycle, VAX RPG II performs total-time calculation and output operations for every cycle.

If control-level indicators are specified, VAX RPG II bypasses total-time calculation and output operations until after the first record with control fields is processed. When the LR indicator is on, VAX RPG II always performs total-time calculation and output operations.

- ⑰ In this step, VAX RPG II performs **all** total-time calculations conditioned by a control-level indicator or containing LO in columns 7 and 8 of the Calculation specification. Total-time calculations can include CHAIN operations, in which a record is immediately retrieved from an input file (see Figure 1-4), or READ operations, in which the **next record** is retrieved from a demand file.
- ⑱ Here, VAX RPG II writes all total-time output lines that satisfy the conditions specified by the indicators. If an **overflow** indicator (OA through OG, or OV) is on, and fetch overflow is specified, VAX RPG II writes the overflow lines as well.
- ⑲ VAX RPG II determines whether the LR indicator is on. If it is **on**, VAX RPG II performs table and array output, closes all files, and terminates the program.
- ⑳ VAX RPG II checks to determine whether any overflow indicators (OA through OG, or OV) are on.

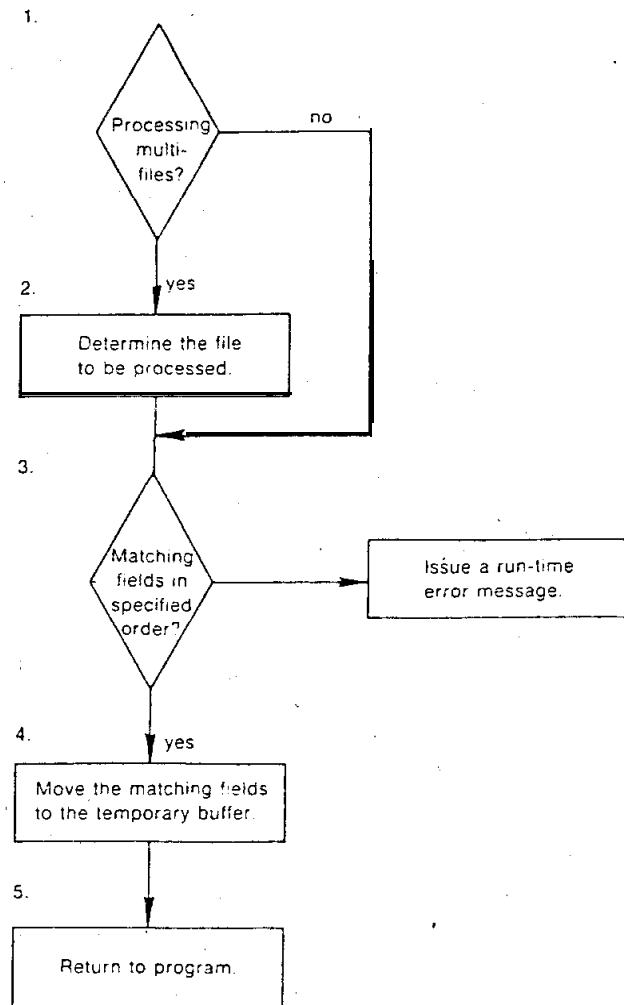
- 21 If any overflow indicators are on, the overflow routine is given control (see Figure 1-5). VAX RPG II outputs all lines conditioned by those overflow indicators that are on. However, VAX RPG II outputs these lines only if they were not output by fetch overflow logic (step 2 or step 18).
- 22 VAX RPG II determines whether the MR indicator should be set on. If this is a **multifile** program and the record being processed is a matching record, VAX RPG II sets the MR indicator on; it remains on for the duration of the cycle during which the matching record is processed. If, these conditions are not present, VAX RPG II sets the MR indicator off.
- 23 If the program contains a **WORKSTN** file, VAX RPG II sets **off** the KA through KZ and KO through K9 indicators. If form input was terminated by a command key, VAX RPG II sets on the corresponding command key indicator. Note that if an error occurred on the read, the command key indicators are not changed.
VAX RPG II extracts data fields from the record to be processed and sets the field indicators on or off, as appropriate, for those fields.
- 24 VAX RPG II then determines whether look-ahead fields are specified in the last file processed and whether it is an input file.
- 25 If the last file processed was an input file with look-ahead fields, VAX RPG II passes control to the VAX RPG II look-ahead routine (see Figure 1-6). In this routine, VAX RPG II retrieves the look-ahead record and extracts the look-ahead fields. If look-ahead fields are not specified, VAX RPG II continues with detail-time calculations
- 26 This is the detail-time calculations step. Here, VAX RPG II performs all conditioned detail-time calculations and subroutines. The calculations may include CHAIN and READ operations (see Figure 1-4). Detail-time calculations complete the VAX RPG II logic cycle. Then, the cycle branches to step 2 to begin again.

ภาพต่อไปนี้จะแสดง RPG II Program Logic Cycle

โดยที่

- ภาพ 1-3 แสดง RPG II matching-field routine
- ภาพ 1-4 แสดง RPG II file processing for chained and demand file
- ภาพ 1-5 แสดง RPG II overflow processing
- ภาพ 1-6 แสดง RPG II look-ahead processing

Figure I-3: Logic Cycle for the Matching-Fields Routine

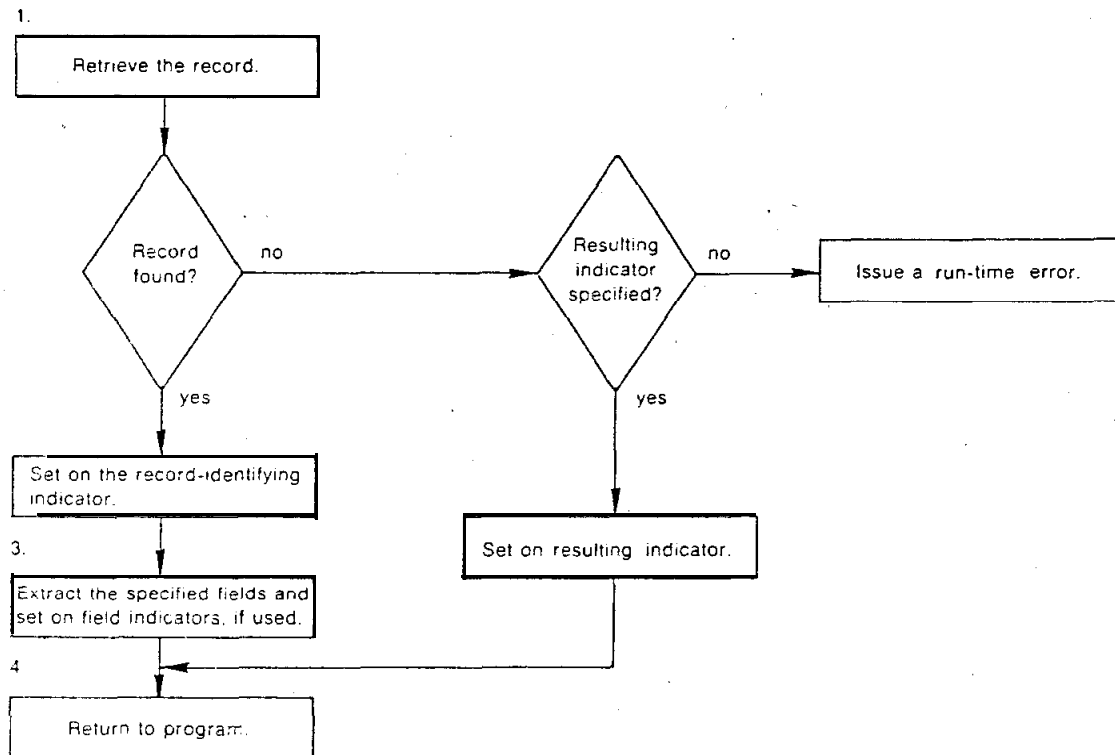


ZK-1159-83

Key to Figure I-3:

- ❶ VAX RPG II determines whether the program uses more than one primary and secondary file. If **multifile** processing is in effect, processing continues with step 2. Otherwise, the program branches to step 3.
- ❷ VAX RPG II compares the matching **fields** to determine which file is to be processed. VAX RPG II extracts the matching fields and checks their **sequence**.
- ❸ If the matching **fields** are not in sequence, a run-time error occurs and the program **terminates**.
- ❹ VAX RPG II moves the matching fields into a temporary buffer. The next record is selected, based on the value of the matching fields.
- ❺ VAX RPG II returns to the program.

Figure I-4: Logic Cycle for Chained and Demand Files

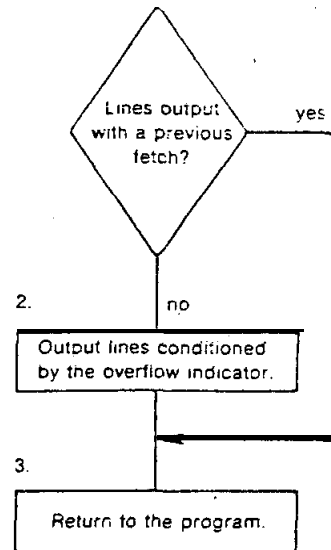


ZK-1459-83

Key to Figure I-4:

- ❶ VAX RPG II retrieves the next record in the file specified by the CHAIN or READ operation code. If the record is not found on a CHAIN operation or an end-of-file occurs during a READ operation and a resulting indicator is not specified, a run-time error occurs. If the record is not found on a CHAIN operation or an end-of-file occurs during a READ operation and a resulting indicator has been specified, the indicator is set on and control returns to the program.
- ❷ VAX RPG II sets on the record-identifying indicator associated with the chained or demand file for the record type read.

Figure I-5: Logic Cycle for Overflow Processing

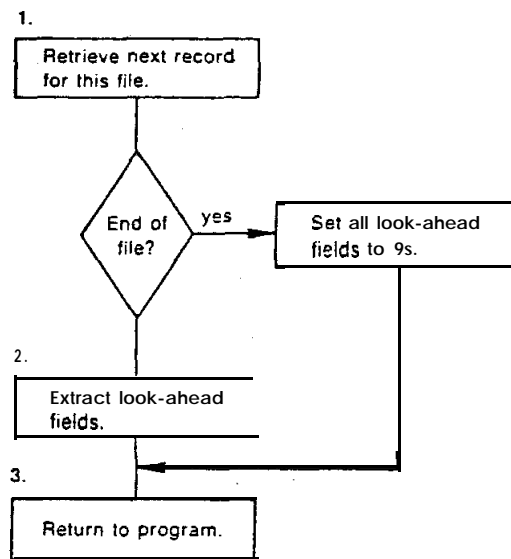


ZK-1460-83

Key to Figure I-5:

- ❶ VAX **RPG** II uses the fetch overflow routine to determine whether the overflow lines were written previously. If the overflow lines were written previously, the program branches to the specified **return** point; otherwise, it continues with step 2.
- ❷ VAX **RPG** II evaluates all overflow lines and writes those lines that satisfy the conditions of the indicators.
- ❸ VAX **RPG** II returns to the program.

Figure I-6: Logic Cycle for Look-Ahead Processing



ZK-1461-83

Key to Figure I-6:

- ① VAX RPG II reads the next record for the file being processed. If the end-of-file **has been** reached, all look-ahead **fields** are filled with 9s and control is returned to the program.
- ⚡ VAX RPG II extracts the look-ahead fields from the record.
- ⌂ VAX RPG II returns to the program.