CORSIM - A DYNAMIC ASSEMBLY LINE SIMULATION

by R. C. Pasternak

Ball Brothers Research Corporation
Boulder, Colorado

ABSTRACT

Corsim is a GPSS program used to experiment with and evaluate changes to an assembly line process for assembling computer memory core planes. The program handles such typical tasks as: two shift operation with independent product assembly but intermixed product test; on the spot transfer of operators to alleviate waiting lines; random defect generation in products with proportionate corrective delays.

Corsim is not a generalized assembly line program, but it does demonstrate techniques which can be applied to similar manufacturing simulations.

INTRODUCTION

Continual changes in the core plane assembly line presented a costly problem to Ball Brothers Research Corporation. A computer simulation of the assembly operation was written to reduce this cost and to allow experimentation with alternate configurations of the process. The program, named CORSIM, was written in IBM's General Purpose Simulation System Language, (GPSS) for the IBM S/360-30 with 64K memory. Detailed in the paper are areas typical of most assembly lines. The purpose of the paper is to exemplify how GPSS is used to perform the simulation.

THE PROGRAMMING SYSTEM

In general, the modest memory size of the computer restricted the number of active transactions (products in stages of assembly) and GPSS entities available to the programmer. Some of these elements were exceeded by more than 200 percent. During normal runs, the simulation required 215 blocks, as many as 600 active transactions, and 136 facilities (stations to service the products). This greatly contrasts the system's allocation of 120 blocks, 200 transactions, and 35 facilities (Figure 1). Many programming techniques can be used to avoid exceeding limits. Usually, however, the governing factor is the number of transactions required to load the model, i.e., to reach a state of real life activity. Several trial runs were required to determine that level in CORSIM. Besides REALLOCATE, other GPSS features such as PREEMPT and MATRIX SAVEVALUE were used to fit the simulation within the system confines.

THE NATURE OF THE ASSEMBLY LINE

The manufacturing of core planes is a highly skilled hand operation. It requires both close attention to minute detail and manual dexterity. The resulting assembly line is not highly automated. Nevertheless, certain problems, such as product testing and movement of personnel, are typical of most assembly lines, automated or not. Other problems may or may not be typical, such as random defect generation and independent shift operation.

PRODUCT AND METHOD

A core plane is a matrix of donut-shaped ferrite loops called cores. The matrix is woven together with three wires, two in the X and Y directions, a third (sense wire) interwoven throughout (Figure 2). These planes make up the memory unit of a computer. In essence, the products pass through several major stages, stations or facilities. In CORSIM these are named as shown below:

1. Matrix load
2. X Y wire installation and test
3. Core replace
4. 3rd wire installation and test
5. Core exchange
6. Finalize

In GPSS, this general flow is represented by STORAGES and FACILITIES, with the appropriate ENTER and LEAVE, SEIZE and RELEASE activities. Each station has its clock advance based on real life. QUEUES are added to receive any waiting lines. For example, a TRANSACTION (product) may SEIZE a testing FACILITY (handles only one product at a time) and advance its manufacturing time by three minutes. After RELEASING the FACILITY, the product may ENTER a STORAGE shelf (handles many products) to await further processing. Eventually, the products LEAVE the shelf and continue in production. Depending on how manufacturing time is calculated, the delay in storage may or may not ADVANCE the clock time. A QUEUE can be entered prior to the testing station, and DEPARTED when that station is available. This flow is shown in Figure 3. (Assume Test Facility is unit 4, and Storage Shelf is unit 9.)

AREAS FOR DETAILED DISCUSSION

A. Product Defects—test and repair

B. Unscheduled Supervisory shift of personnel

C. Batch processing

D. Day/Night shift simulation

PRODUCT DEFECTS—TEST AND REPAIR

Because of their significant contribution to time delay, the process of testing and repairing defects is usually essential to any simulation. In CORSIM, there are two major tests for defects as follows:

1. X Y tests—made immediately after the X Y wires are installed to determine whether the cores at the wire intercepts are operative. Retesting after repair does not improve the process or product.

2. 3rd wire test—made immediately after the sense wire is installed. Similar to X Y test except that products are recycled until all defects are corrected.

There are several types of defects; rates are naturally high compared to simpler products

Defect generation was built into the simulation for each testing station. A statistical analysis provided the mean and standard deviation for each defect type by test and by shift (Figure 4). Curves were prepared using the GPSS FUNCTION feature. A GPSS random number generator selected a point along the appropriate function curve and automatically a number of defects were determined for a given core plane. Repair time was calculated as a GPSS VARIABLE directly related to the defect assignment. Figures 5 and 6 are normally distributed function curves prepared by IBM.[1] Figure 7 represents a type of defect distribution used in CORSIM. Numbers are not actual.

UNSCHEDULED SUPERVISORY SHIFT OF PERSONNEL

In a series type assembly line, the rate of activity at each stage is directly dependent on the preceding step; key elements are monitored carefully. A supervisor will often shift personnel from some operations to alleviate bottle necks in others. In the Core plane assembly line, "core replace" was a critical facility. Failure to program this optional personnel shift into early simulation runs caused large queues at "core replace" and zero utilization of personnel further down the line. A revised model tested the utilization of facilities dependent upon "core replace" activity. At specified low limits, additional "core replace" facilities were generated and a corresponding number of facilities were suspended in other non-essential operations. The situation was returned to normal when the utilization of the dependent activities reached an upper limit. This may seem to destroy the value of queuing. But in this instance, the basis for establishing the true queue was the efficient utilization of dependent facilities. For this reason, we tested utilization of other facilities rather than queue size at core replace. The latter would have resulted in a zero queue at core replace. The GPSS standard numerical attribute "WAIT" coupled with GATE on LOGIC SWITCH were used for this maneuver. To cut down on the number of transactions in queues, a quantity of defective products in excess of a specified limit was stored in a MATRIX SAVEVALUE. This tactic kept the model within the system confines. In keeping with the actual assembly line procedure, certain of these transactions were shifted via LOGIC SWITCHES to night shift operations.

BATCH OPERATIONS

Batch or lot operations are typical of many assembly lines. In the core plane system, cores (raw material) were received in batch shipments. A given batch had certain characteristics and was used to manufacture a set of core planes. The number of core planes in a set obviously varied as the size of a batch. Averages and standard deviations helped produce a function curve which directly determined the distribution of core planes by batch allocation. Batch numbers were controlled through SAVEVALUES. These and the total number of core planes by batch were assigned as TRANSACTION PARAMETERS. The shipping facility used these parameters (much as a traveller sheet) to determine when a batch was completed.

DAY/NIGHT SHIFT OPERATIONS

It is normally quite easy to simulate the operation of changing from day to night shift. However, in CORSIM this was perhaps one of the most complicated phases of the simulation for the following reasons:

1. There were different numbers of people at each station.

[1] GPSS/360 User's Manual, IBM, pp. 31, 33

2. Core planes being worked during the one shift were stored at the end of that shift. Work on these items was not resumed by the next shift. In effect, each shift had its own products (except for those noted above).

3. Certain operations performed at night were not done during the day.

Therefore, at change of shifts the objective of the program was to suspend shift operations, store the products in whatever state of assembly, initiate the next shift with its respective products partially complete from yesterday's work, and activate new facilities. Several GPSS features were used to accomplish the nitty-gritty of shift changing:

1. GENERATE—used to produce the proper shift intervals.

2. TEST—used to determine night or day time period.

3. GATE on LOGIC SWITCH—used to select day or night conditions.

4. PREEMPT and RETURN—used to suspend and reactivate the work of each shift.

5. SPLIT—used to continue product flow while shift change was in progress.

SUMMARY

An assembly line simulation has been presented. Several program details were explained to demonstrate how the simulation was accomplished. Hopefully the paper showed that GPSS fulfilled the modeling requirements. Applications to similar problems in industry can be inferred by the reader.

Entity Type	Basic Core Allocation Per Item (bytes)	64K Quantity Allocation	CORSIM Requirements
Transactions	16*	200	600
Blocks	12	120	215
Facilities	28	35	136
Storages	40	35	3
Queues	32	70	7
Log Switches	6	200	9
Tables	48	15	1
Functions	32	20	5
Variables	48	20	3
Savevalues (fullword)	4	100	10
Savevalues (halfword)	2	50	-
User Chains	24	20	-
Groups	4	5	-
Boolean Variable	32	5	1
Matrix Savevalue (full)	24	5	2
Matrix Savevalue (half)	24	5	-

*Add 20 bytes of common for every active transaction plus additional words for parameters.

Figure 1 GPSS/360 ENTITIES AND CORSIM REQUIREMENTS

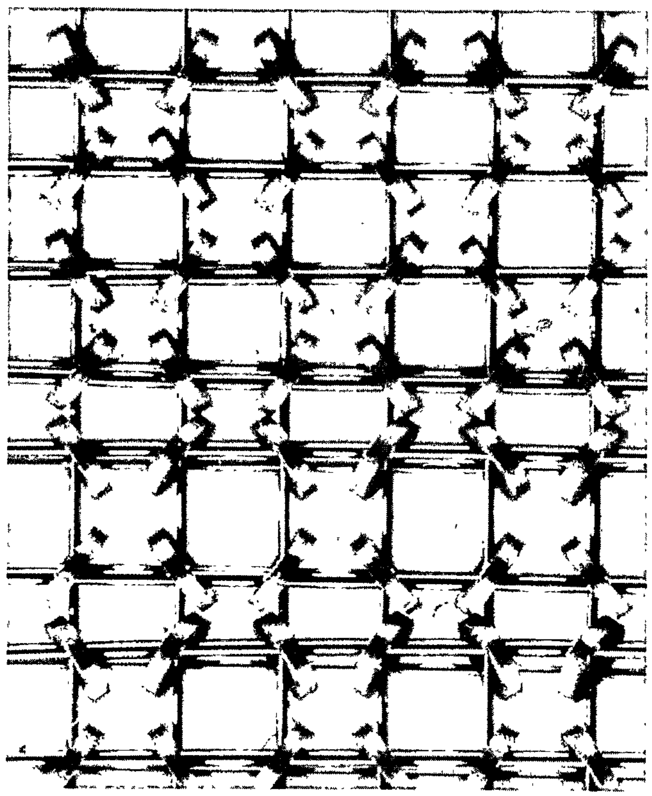

Fig. 2 Core Plane Closeup

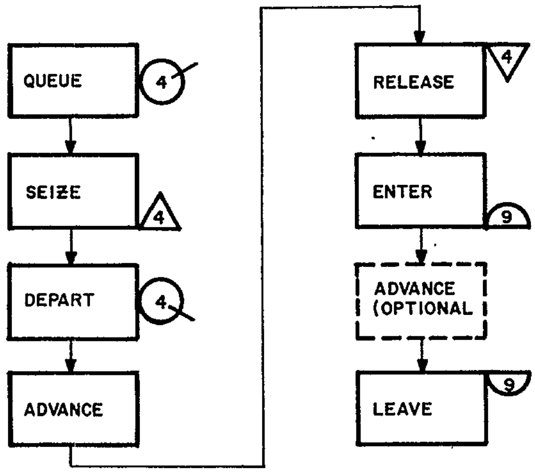

Fig. 3 Flow Chart

```
|2   |8          |19
|1   |FUNCTION   |RN1, C24
```

1	7	13	19	25	31	37	43	49	55	61	67
0	0	0.1	0.104	0.2	0.222	0.3	0.355	0.4	0.509	0.5	0.69
0.6	0.915	0.7	1.2	0.75	1.38	0.8	1.6	0.84	1.83	0.88	2.12
0.9	2.3	0.92	2.52	0.94	2.81	0.95	2.99	0.96	3.2	0.97	3.5
0.98	3.9	0.99	4.6	0.995	5.3	0.998	6.2	0.999	7	0.9998	8

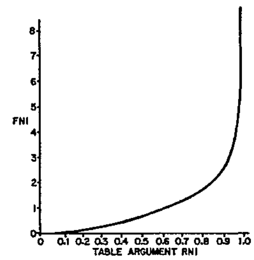

Fig. 6 Function Curve

DAILY STATISTICS FOR SHIFTS

SHIFT NO. 1

TOTAL DEFECTS CODE (1)
MOTH HOLE DEFECTS CODE (2)
RV1 DEFECTS CODE (3)
WVZ DEFECTS CODE (4)

SHIFT 1 SHOP DATE 0
SHIFT 2 SHOP DATE 0

	TOTAL	AVG	SD	VMIN	VMAX	DELETE
TEST 1						
DEFECT CODE 1	168	6.91	4.07	0.0	18.00	0
DEFECT CODE 2	168	6.66	5.43	0.0	27.00	0
DEFECT CODE 3	168	6.91	4.07	0.0	18.00	0
DEFECT CODE 4	168	0.0	0.0	0.0	0.0	0
TEST 2						
DEFECT CODE 1	64	0.97	1.95	0.0	12.00	0
DEFECT CODE 2	64	0.0	0.0	0.0	0.0	0
DEFECT CODE 3	64	0.86	1.94	0.0	12.00	0
DEFECT CODE 4	64	0.02	0.13	0.0	1.00	0
TEST 3						
DEFECT CODE 1	122	0.12	0.40	0.0	2.00	0
DEFECT CODE 2	122	0.0	0.0	0.0	0.0	0
DEFECT CODE 3	122	0.10	0.37	0.0	2.00	0
DEFECT CODE 4	122	0.0	0.0	0.0	0.0	0

Figure 4 Representative Statistics for Defect Distribution

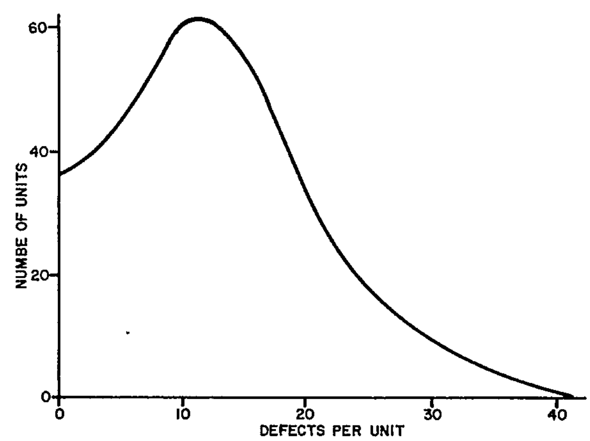

Fig. 7 Representation of Defect Distribution

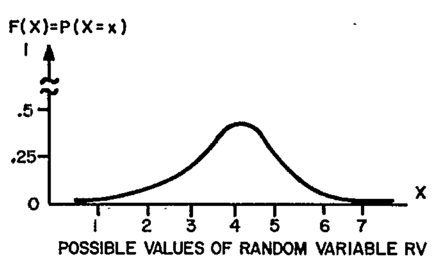

Fig. 5 Probability Density Function