Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, October 25-26, 2012

Schedulability Analysis of SCOPS on a Platform for Safety I&C Systems of SMART MMIS

Jong Yong Keum[a*], Yong Suk Suh[a], Kwang-Il Jeong [a], Yong Jin Seo[b], Hyeon Soo Kim[b], Je Yun Park[a]
[a]*Research Reactor Design & Engineering Div. KAERI, 150-1 Dukjin-dong, Yuseong-gu, Daejon, Korea, 305-353*
[b]*Department of Computer Science and Engineering, Chungnam Nat'l Univ., 220 Gung-dong, Yuseong-gu, Daejon, Korea, 305-764*
[*]*Corresponding author: jykeum@kaeri.re.kr*

1. Introduction

A real-time I&C system used in safety systems in nuclear power plants shall have predictable and deterministic characteristics. The main issue of predictable real-time system is to prove whether it satisfies its deadline. One way to prove whether a real-time I&C system satisfies its deadline is a schedulability analysis. A schedulability analysis on SCOPS (SMART COre Protection System) is performed.

2. Design Characteristics of the Scheduler and Data Communication

2.1 Tasks Run in a Deterministic Sequence

The fixed execution orders of the tasks at the design phase are predefined. Based on the predefined execution orders, execution intervals, and execution time of the tasks, it is demonstrated that the tasks are run in a deterministic sequence.

2.2 Single Task Architecture and No Interrupt

The tasks in a single task architecture do not compete to occupy resources. After a task completely finishes its execution, the next task starts its execution. The running task occupies all resources required. When it terminates, it frees up the resources. There is no interrupt except for a timer interrupt. The processing period of the scheduler is 25ms. The scheduler architecture is shown in Fig. 1.

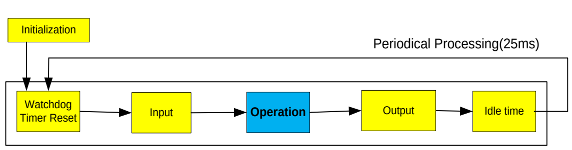

Fig. 1. The Scheduler Architecture

2.3 No Input/Output Blocking and No Deadlock

When a running task waits for the required resources, we refer the situation as blocking. The blocking is divided into two cases. The first is to wait for the input/output. The other is a deadlock. Input/output blocking is used to wait for the reading or writing of data to global buffers or communication buffers to input or output the data. The communication buffers use dual-ported RAM, which is a type of random access memory that allows multiple reads or writes to occur at the same time. As mentioned above, because a scheduler has a single task architecture, a situation in which a task waits for resources occupied by other tasks does not occur. The same reason is also applied to a deadlock.

2.4 Deterministic Data Communication

To ensure deterministic communication, the state-based transmission method is adopted [3]. The data communication load is designed to be constant by this method. The data communication protocol does not use handshaking, which provides a nondeterministic property.

3. Test Results

To analyze the schedulability of SCOPS, the execution time and execution order for each task, the pre-processing time for the initialization, and extra time, which is time difference between the termination of any task and starting of another task are required.

3.1 Execution Times of Tasks

To show that the scheduler ensures schedulability, the execution time per task is measured using an oscillator. The execution period [4] and execution order of a task are shown in Table 1.

Table 1. Execution period and order of a task

Tasks	Execution Period	Execution Order
Watchdog Timer Reset	25ms	1
Input	50ms	2
CRAPOS	100ms	3
CORVAR	50ms	4
TRPGEN	50ms	5
CORAPD	250ms	6
STDNBR	2000ms	7
Output	50ms	8

The execution time of each task is shown in Table 2. The execution times according to the execution paths are shown in Table 3. The longest execution path is execution path 6, which is executed per 2000 ms. The execution time of the path, which is the worst case execution time, is

14.44 ms. Therefore, it is concluded that the real-time scheduler can ensure the schedulability of all tasks.

Table 2. Execution times of tasks

Task Module		INPUT	CORVAR	TRPGEN	CRAPOS	CORAPD	STDNBR	OUTPUT
Exec. Period (ms)		25	50	50	100	250	2000	25
Exec. Time (ms)	Average	0.205621	0.006615	0.001239	0.056575	0.147811	13.855560	0.162426
	Maximum	0.206067	0.006707	0.001253	0.056853	0.148133	13.855627	0.162867

Table 3. Execution time of execution path

Exec. Path	Execution Task						Exec. Time (ms)	
							Aver.	Max
1 (25ms)	INPUT	CORVAR + TRPGEN	CRAPOS	CORAPD	STDNBR	OUTPUT	0.368047	0.368934
2 (50ms)	INPUT	CORVAR + TRPGEN	CRAPOS	CORAPD	STDNBR	OUTPUT	0.375901	0.376894
3 (100ms)	INPUT	CORVAR + TRPGEN	CRAPOS	CORAPD	STDNBR	OUTPUT	0.432476	0.433747
4 (250ms)	INPUT	CORVAR + TRPGEN	CRAPOS	CORAPD	STDNBR	OUTPUT	0.523712	0.525027
5 (250ms)	INPUT	CORVAR + TRPGEN	CRAPOS	CORAPD	STDNBR	OUTPUT	0.580287	0.581880
6 (2000ms)	INPUT	CORVAR + TRPGEN	CRAPOS	CORAPD	STDNBR	OUTPUT	14.435847	14.437507

3.2 Pre-processing time and Extra Times of Execution Paths

Because the scheduling method is non-preemptive, the context time switching time for any task is not considered. However, there is extra time which is the time difference between the termination of any task and the start of another task. Fig.2 shows the conceptual pre-processing time and extra time.

Extra times according to execution paths are shown in Table 4. The execution time of execution path 6 as the worst case execution time is 0.00068 ms. The pre-processing time is about 0.0004-0.0005 ms. Therefore, the summation of the pre-processing time and extra time is 0.00118ms.

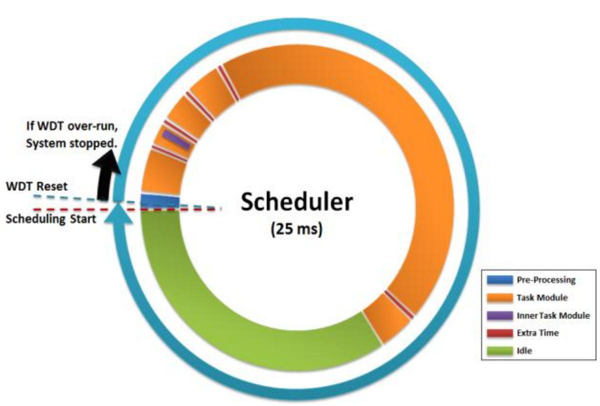

Fig. 2. Conceptual diagram of pre-processing and extra time

3.3 Execution time of input task according to data transmission time interval

Table 4. Extra time according to the execution path

Execution Path	1 (25 ms)	2 (50 ms)	3 (100 ms)	4 (250 ms)	5 (250 ms)	6 (2000 ms)
Extra Time (ms)	0.0003	0.00038	0.000461	0.000467	0.000547	0.00068

The input task reads all data in the buffers of the communication board until they are empty. The more data in the buffers, the more execution time of the input task required. To verify this phenomenon, a test measuring the execution time of the input task according to the data transmission time interval was performed. The test results are shown in Table 5.

Table 5. Execution time of input task according to data transmission time interval

Data Transmission Period	Execution Time of INPUT Module (ms)				
	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
25 ms	0.2056	0.2061	0.2065	0.2060	0.2056
20 ms	0.206	0.4092	0.2064	0.2064	0.2064
10 ms	0.6121	0.409	0.6121	0.613	0.4091
5 ms	0.8159	1.02	1.018	1.02	1.018
2 ms	2.44	2.44	2.44	2.64	2.64
1 ms	5.081	4.87	5.081	5.079	4.882
500 us	9.95	10.16	10.36	10.36	9.95

As shown in Table 5, the shorter the data transmission interval, the longer the execution time of the input task. These test results can be considered to determine the data transmission time.

4. Conclusions

To show that all tasks of SCOPS meet their deadlines, a schedulability analysis was performed. When the processing period of the scheduler is 25 ms, the execution time of the longest execution path is 14.44 ms. This satisfies the deadlines of all tasks of the SCOPS. Through these test results, it can be shown that predictable and deterministic characteristics for safety I&C systems of SMART MMIS were implemented.

REFERENCES

[1] Kwang-Il Jeong, et al, Communication board design to satisfy the communication independence in safety I&C system, Transactions of the Korean Nuclear Society Spring Meeting, Jeju, Korea, May 17-18, pp.1178-1179, 2012

[2] Korea Atomic Energy Research Institute, 004-NR444-001, Rev.01, SMART SCOPS Functional Design Requirements