AN EFFICIENT METHOD FOR ENHANCING THE QUALITY FOR DANGER CONTROL SYSTEMS

Corneliu Popescu

University of Oradea, Str.Armatei Române nr.5, Oradea, România
Corneliu_popescu@uoradea.ro

Abstract: A danger control system is designed for managing different building dangers and any specific building activity. The danger control systems are not productive systems, they are responsible for creating the preconditions for a reliable functioning of any type of building. So, the danger control systems are an essential prerequisite for the reliable and efficient functioning of the life-safety measures and security in a building [2]. Sometime, even the existence of such a building, the people life, the goods integrity and the activity carried on is dependent of the danger control system's quality. The present paper give a potential solution for enhancing the quality for danger control systems.

Key words: danger control system, risk sectors, programmable output, testability, quality

1. INTRODUCTION

A danger control system is designated especially for the detection of and response to danger incidents. The most important function of a danger control system is to provide the user with clear information on critical situations so that he has no doubt as to what counter measures to initiate. Particularly critical is the occurrence of multiple alarms and messages in hectic situations: the operator should know precisely what to do in every phase. A danger control system can be defined as a danger state diagnose and control system because it must detect, localize and then control the danger event. The analyze of "potential dangers" at which a building is exposed permits a classification of dangers in "risk sectors" [2]: Fire, Extinguishing, Intrusion, Gas, and Building services. The risk sectors are mainly defined by the kind of detector sensor, by their signal display, operation and control functions and by their alarm organization. The control units are responsible for the management of the danger events specific to any risk sector [2]. This paper presents an efficient method with practical applicability that can be used for enhancing the testability of such systems in conformity with the needed requests.

2. THE CONTROL UNIT AS THE MAIN ELEMENT OF DANGER CONTROL SYSTEM QUALITY

The danger control system quality is dependent on system availability [7] which can be positively influenced by specific measures for enhancing the system reliability

and / or testability. The present paper focuses on enhancing the quality by enhancing system testability at control units (CU) level. Fig.1. presents the level structure of a danger control system.

The control unit makes in principal the fault diagnose for faults outside the control unit [3] [4] [6], for module connected to its bus, for alarm voltages and alarm devices. So, in Fig.2:

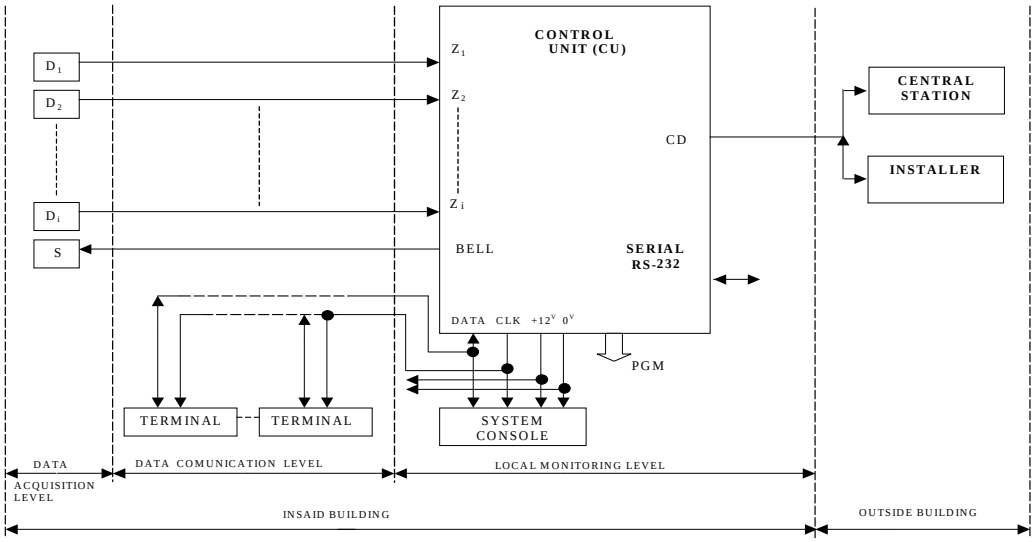

Fig.1.

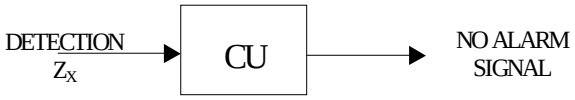

Fig. 2.

we have a controller unit fault that practically is not monitored. This aspect is valid for a very large range of actual generation control units. Today, this kind of fault is detected by switching in the test mode with the *WALK TEST INSTALLER* function; but as we have already explained [7] this aspect is highly time consuming and reduces the danger control system availability.

Generally, the testing methods are classified as being on-line methods and off-line methods. Sedmak [1] has proposed a self-verification method, which is a combining trial of the two methods. Sedmak defines the self-verification as being an automated verification logic for functioning without faults that eliminates the need for applying external stimuli (others than clocks and power supply)

It is important to note that each zone, conforming to manner it was defined ([4] – zone definitions), is independently handled by the control unit. So, as it is shown in Fig.3, we can say that the smallest unit of control for a control unit is the zone.

Based on the danger control system requirements [2] [7] and on the aspects that we have already explained the conclusion is that the test is better to be done on-line, during the normal functioning of the control system. This requirement make impossible the test pattern generation, because the generation of some test stimulus (during the normal functioning) on the zone input can generate at different moments the triggering of some

unpleasant alarms. On the other hand, the use of a test pattern generator implies some extra costs and the choice of some appropriate moments for running the test sequence. I tried to find an appropriate test for covering this kind of faults at the control unit level; that is to test the control unit controller in a manner similar with the *WALK TEST INSTALLER* function (by verifying the event buffer) and that will be executed without switching in the special test mode that I have already mentioned.

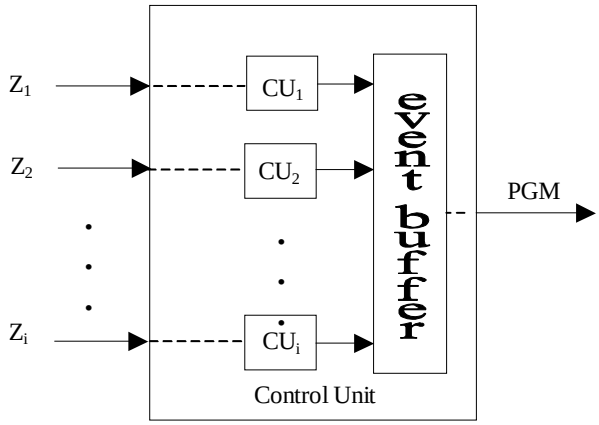

Fig.3.

3. THE PGM OUTPUTS USED AS AN ENHANCEMENT TESTABILITY SOLUTION FOR DANGER CONTROL SYSTEMS

The control units have some special outputs called *Programmable Output Module* [3][4][5][6]. There are many possibilities for programming them (24), but I chose for programming them with the definition 10 - *Latched System Event* (*Strobe Output*). This programming mode permits to program PGM outputs to activate when special events occur. These events can be programmed in the following manner:

[1]-- *Burglary* (*Delay, Instant, Interior*)
[2]-- *Fire* (*Fire Keys, Fire Zones*)
[3]-- *Panic* (*Panic Zones*)
[4]-- *Medical* (*Medical and Emergency Zones*)
[5]-- *Supervisory* (*Supervisory, Freezer and Water Zon*es)
[6]-- *Priority* (*Gas, Heat, Sprinkler*)
[7]-- *Holdup* (*Holdup Zones*)

Practically, the events received at the zone inputs, are registered in the CU event buffer, and then, they can be transmitted (by an appropriate programming) to the programmable output, that can be used in the testing process. It is important to note that a major problem for the above mentioned method is the fact that after the event occurring, the PGM output so defined will remain stuck-at 1 until we reset it with a reset cod from the console [3] [4] [6]. This is because the CU producers did not provide this definition mode (10) for PGM output for testability reasons; it was introduce for producing some interactions inside control system. On the other hand is important to maintain the current control unit state after reset. Globally, the control unit may have 2 states:

- Armed

- Disarmed

I tried to find another way then that provided by the CU producers, for resetting PGM outputs defined as *Latched System Event;* the two majors requests for this new way are:

- Automatic reset for a PGM output defined as *Latched System Event,* without the operator need
- Keeping the current CU state even after the reset

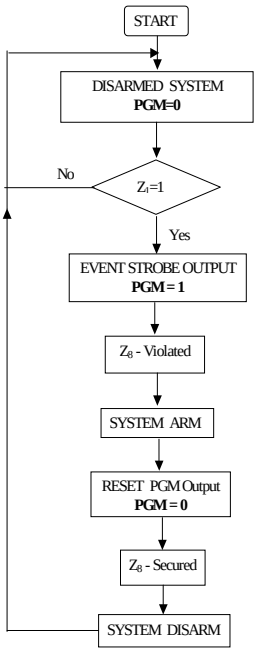

Fig.4.

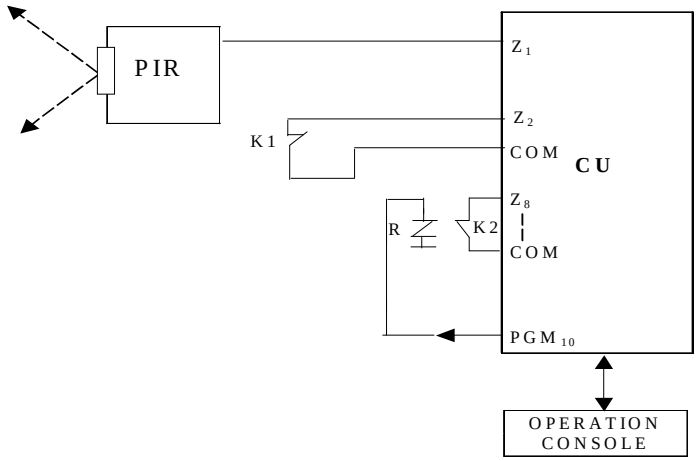

Fig.5.

So, I considered the (23) PGM output definition - *Maintained Keyswitch Arm Zone* [18], that provides successively CU arm / disarm states when this zone is momentary violated / secured. The definition permits me to conclude that at the moment when a zone defined with (23) definition is violated / secured, the arm cod is automatically transmitted on the data internal line of the controller (CU).

For testing the solution, I used 2 CU zones and one PGM output. Z1 zone is defined as 24 hours zone; this meaning a zone when the produced events are recorded to the PGMx output, defined as *Latched System Event* (10). Z8 zone is defined by "*Maintained Keyswitch*" definition and it has the role to reset the event and to bring it in the initial state. The control unit successive states are shown in Fig.4, and the implementation for the automatic reset is given in Fig.5: the events are produced on Z1 zone by a motion sensor (passive infrared – PIR).

A PC 5010 [4][5] was programmed for testing and the results are the following:

```
$Table=Zone 1 to 8 Definitions [001]
Zone      Definition
Zone 01   (15) 24 Hour Medical
Zone 02   (22) Momentary Keyswitch Arm
Zone 03   (14) 24 Hour Heat
Zone 04   (14) 24 Hour Heat
Zone 05   (14) 24 Hour Heat
Zone 06   (14) 24 Hour Heat
Zone 07   (00) Null Zone (Not Used)
```

```
Zone 08   (23) Maintained Keyswitch Arm


$Table=PC5010 PGM's 1 & 2 (Onboard) [009]
Attributes,Definition
PGM 01   (19) (*71) Command Output #1
PGM 02   (10) Latched System Event (Strobe)
```

I induced events on Z1 zone both for the armed CU state and for the disarmed CU state. The testing results are given by the event buffer contain, which is listed below.

```
$Table=Event Buffer
System     07.03.2002  03:02   Installer Lead Out
System     07.03.2002  03:01   Installer Lead In
System     07.03.2002  03:00   Special Opening
System     07.03.2002  03:00   Opening by Keyswitch Zone
System     07.03.2002  03:00   Special Closing
System     07.03.2002  03:00   Closing by Keyswitch Zone
System     07.03.2002  03:00   Opening After Alarm
System     07.03.2002  03:00   Special Opening
System     07.03.2002  03:00   Opening by Keyswitch Zone
System     07.03.2002  03:00   Alarm Restore Zone 1 - EVENI
System     07.03.2002  03:00   Recent Closing
System     07.03.2002  03:00   Alarm Zone 1 - EVENI
System     07.03.2002  03:00   Special Closing
System     07.03.2002  03:00   Closing by Keyswitch Zone
System     07.03.2002  03:00   Special Opening
System     07.03.2002  03:00   Opening by Keyswitch Zone
System     07.03.2002  03:00   Special Closing
System     07.03.2002  03:00   Closing by Keyswitch Zone
System     07.03.2002  03:00   Alarm Restore Zone 1 - EVENI
System     07.03.2002  03:00   Alarm Zone 1 - EVENI
System     07.03.2002  03:00   Special Opening
System     07.03.2002  03:00   Opening by Keyswitch Zone
System     07.03.2002  03:00   Special Closing
System     07.03.2002  03:00   Closing by Keyswitch Zone
System     07.03.2002  03:00   Alarm Restore Zone 1 - EVENI
System     07.03.2002  03:00   Alarm Zone 1 - EVENI
System     07.03.2002  02:59   Alarm Restore Zone 1 - EVENI
```

As we see from the event buffer, the solution was tested for both the armed and disarmed initial states, and for both situations we succeeded to reset PGM output and to maintain the system state.

4. TESTABILITY DEVELOPMENT POSSIBILITIES BASED ON THE PROPOSED SOLUTION

To be noted that this solution permits to obtain test vectors automatically, without the need of an extra hardware. It is sufficient the existing hardware. Moreover, it easily to be seen the special flexibility provided by these control units in tracing the various system events. As an example, defining a PGM output by (09) definition - *System Trouble Output*, this will be activated / deactivated on occurrence / loss of fault conditions mentioned in [4].

Based on the arguments pointed out, on the PGM output flexibility, I assume that there are enough conditions to conclude that based on different test criteria (C_{Ti} , i =1,….k) and on the needed testability degree for a given application, we can make k groups of test vectors C_{Ti}. Fig.6 illustrates a potential use of the adopted solution by a redundant structure, static, global with voting and self-test. In this manner, the danger control system's testability degree may be spectacularly extended based on a large number of criteria. By example we can make up test vector groups referring to the risk sectors of a security system: intrusion, fire extinguishing, gas lacks, flood, panic,

medical, holdup. There are also some other criteria that may take part in making up test vectors according the testability imposed requirements.

5. CONCLUSIONS

A new solution is promoted by this paper (contrasting with [3] [4] [6]), that allows the automatic reset of the control units PGM outputs, offering the possibility for tracing the system events based on programmable outputs. So, the condition of maximizing the system observability which determines the enhancing of the control system testability. This solution doesn't require an extra hardware for generating test vectors, these being generated by danger control system detection elements themselves, during normal functioning. To be noted the modularity and flexibility of the solution that permits its expansion by simple repeatability according to high testability requirements, based on different criteria directly imposed by the implemented application, with direct involvement in danger control system enhancing quality.

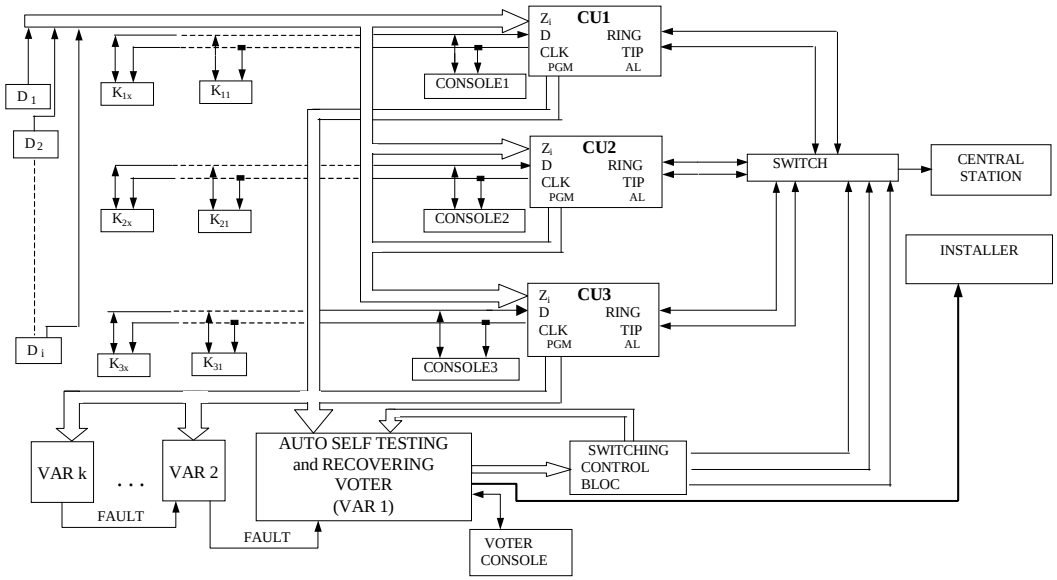

Fig.6.

REFERENCES:

1. R.M.Sedmark, (1980), Design for self-verification: An approach for dealing with testability problems in VLSI-based designs, *Proc.1980, IEEE Test Conf*., pp. 112-120

2. Cerberus AG (1992), Danger Management System, Switzerland, *white Papers,* pp.III.3-III.25

3. Napco Security Systems, Inc. (1996), Gem-P3200, *Control Panel – Programming Instructions*, New York

4. Digital Security Controls Ltd. (1998), P*anel Control PC5010 – Installation Manual*, Canada

5. Digital Security Controls Ltd. (1998), Panel Control PC5010 – *Programming Worksheets*, Canada

6. Digital Security Controls Ltd. (2000), Panel Control PC5020 – *Installation Manual*, Canada

7. Popescu, Corneliu, (2001), Contribu ii privind cre terea fiabilit ii i testabilit ii sistemelor de securitate, *Tez de doctorat, Timi oara*, pp.I.1-I.40