Microprocessors and Microsystems 39 (2015) 435–443

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Formal approach for the safety assessment of embedded controller based on programmable electronic hardware

Jaspal S. Sagoo *

QinetiQ, St Andrews Road, Malvern, Worcestershire WR14 3PS, United Kingdom

ARTICLE INFO

Article history:
Available online 2 July 2015

Keywords:
Petri nets
Safety assessment
Hardware assurance
Previously developed hardware
Hardware standards
Certification

ABSTRACT

The issue of providing assurance for programmable electronic hardware (PEH) that have either been previously developed or composed of Commercial-Of-The-Shelf (COTS) and used in embedded control systems is examined. Specifically, these type of PEH are difficult to assure because no evidence may be available on their development and limited functional descriptions may exist to perform a safety assessment. This problem is addressed by presenting a formal approach that allows a safety assessment on a PEH to be performed. This approach uses a system's architecture and mechanisms such as safety nets to deduce the behaviour of the PEH, which is then translated into the formalism of Petri nets. Since this formalism can be used to model both faulty and non-faulty behaviour, it allows a safety assessment to be performed. Application of this approach is shown via a case study in which a safety assessment is performed for a PEH based embedded controller for an engine control application.

1. Introduction

Due to the availability of powerful and relatively low-cost programmable electronic hardware (PEH), devices such as Field Programmable Gate Arrays (FPGAs) are increasingly being used in safety critical applications (such as avionics, automotive and medical systems) [1–3]. For these applications, the system developer not only has to ensure that the PEH satisfies its functional requirements but also, as required in aviation [4], may need to provide assurances on its safety to a certification body. An acceptable means of providing these assurances is via a safety case [5–8]. Typically, the safety case argument uses evidence such as compliance of PEH development to an accepted standard (e.g. DO 254 [9] and IEC 61508 [10,11]) and safety assessments which show the mitigation of PEH related hazards. Whilst assurances can be readily provided for 'new' FPGA developments, issues may occur for PEH such as Commercial-Of-The-Shelf (COTS) or previously developed hardware (PDH). In these cases insufficient development evidence and limited functional descriptions may prevent the formation of a safety argument. This paper addresses this issue by proposing an assessment approach that can be used to produce supporting evidence for a safety case for a PDH based system.

The guidance of [9,11] shows that (in addition to performing overall system verification) evidence such as service experience, Electronic Component Management plans, safety and impact change analysis may be used for the certification of COTS PEH and PDH. However, it is not clear what approach should be used for systems that do not have such evidence. Rigorous system testing does provide key evidence to show that a system satisfies its requirements under fault and non-fault conditions and can be used to support a safety case. However, it is cost-intensive and may not reveal all failures in the system.

Current PEH standards [9,12] show that techniques such as safety specific analysis and formal methods can be used to perform safety and failure analysis of PEH based systems. Based on this guidance, this paper presents an approach that uses formal analysis to perform a safety assessment on a PDH based system and considers its use in a safety case. This approach uses a case study which concerns a PEH based embedded controller for an engine control application (which is typically used within an avionics or automotive application). The embedded controller uses PDH and its overall functionality is known, however, no evidence exists on how it was developed.

The PEH assessment approach is based on integrating existing techniques to form a systematic method for performing a safety assessment. Specifically, since limited information is known about the internal operation of the PEH, the overall system architecture is examined to determine the PEH behaviour. The information flow related to the PEH and use of safety nets [13] in the system architecture, which monitor or detect anomalous PEH behaviour, can provide important clues to PEH functionality. The PEH behaviour

* Tel.: +44 (0)1684 895188 (work), +44 (0)7767005854 (mobile).
E-mail address: jsagoo@qinetiq.com

http://dx.doi.org/10.1016/j.micpro.2015.06.012
0141-9331/© 2015 Elsevier B.V. All rights reserved.

is modelled using Petri nets [14–16]. This formalism is considered because it has been widely used to analyse concurrent systems comprising software and hardware [17] and is supported by a range of modelling tools. More significantly, it is recommended in IEC 61508 [10] as one of the modelling techniques for performing failures analysis, and can qualify as a formal method to perform safety specific analysis in DO 254 [9]. In this paper the generation and analysis of the Petri net model is performed using CPN Tools [18]. The safety assessment is performed by examining the state reachability graph of the Petri net model under fault-free and fault conditions.

This paper is organised as follows. Section 2 provides a review related work and a description of the safety techniques that can be used to perform the safety assessment of FPGA based systems. Section 3 provides the application context and an analysis of the system. The proposed safety assessment and its evaluation are documented in Sections 4 and 5. Section 5 provides a conclusion on this paper.

2. Related work

Techniques such as Hazards and Operability Study (HAZOPS), Fault Tree Analysis (FTA) and Failure Mode and Effects Analysis (FMEA) are widely used within various industrial sectors to perform safety analysis throughout a product's lifecycle [19,20]. HAZOPS is applied in a top-down manner to identify the system hazards by using a set of guide-words to determine the deviations from the design-intent. FTA uses system hazards, possibly from a HAZOPS, to identify system components or system functions that can cause a hazard. The FMEA is applied in a bottom-up manner to determine the effect of component failure modes on the overall system. The above techniques can be applied either separately or together and at different levels of abstraction (such as system, top-level or detailed design) to perform a system safety assessment [21]. However, since they require a detailed description of a system (i.e. component functionality and interconnectivity) to perform analysis, they may be of limited use for PDH and PEH COTS based systems.

Research in techniques for analysing FPGAs for safety critical applications have typically focused on examining the impact of Single Event Effects (SEEs) [22,23]. The work of [22] proposes a semi-automated analysis techniques (which is based on a FMEA) that can be used to trace low-level SEEs generated FPGA faults to high-level system hazards. The effect of SEEs is noted on the system via synchronisation, propagation and timing issues. Although this analysis seems to provide a valuable PEH safety assessment, as a prerequisite, it requires a VHDL netlist. However, such detailed design data may not be available a COTS PEH or PDH.

The work of [23] propose a technique for identifying those aspects of a System-on-Chip (SoC) that are susceptible to soft errors, so that they could be protected using fault-tolerant design techniques. This technique is based on developing fault hypotheses and risk models (which uses a FMEA like approach) on a SoC represented using SystemC. This approach provides a method of identifying vulnerable components in a SoC by assigning them a failure rate (based on IEC 61508), safety integrity level (SIL) and risk priority number. Since this method requires detailed descriptions of PEH in SystemC, it is difficult to apply to COTS PEH or PDH.

3. Concept of a safety net

The Federal Aviation Administration (FAA) guidance of [13] recognises that due to the highly integrated, complex and nondeterministic nature of PEH and software in aircraft systems there is increasing difficulties and costs of design assurance for these components. Moreover, for these types of systems, it may not be feasible to show that complex aircraft systems are sufficiently free of anomalous behaviour by evaluating system components. Hence, the FAA guidance of [13] suggests the use of safety nets as an alternative approach for mitigating unforeseen or undesirable COTS microprocessor operation by detecting and recovering from anomalous behaviour at the operational system level.

In [13], a safety net is defined as 'the employment of mitigations and protections at the appropriate level of aircraft and system design in order to help ensure continuous safety flight and landing'. In terms of architectural mechanisms, safety nets detect for unexpected changes in behaviour or physical characteristics and can be represented by:

- Hardware monitors that use hardware-based technology for implementing checks such as parity, error correction codes, SEE monitors (for memory, internal and external busses) and dissimilar hardware.
- Software monitors that are implemented in software for performing checks such as data integrity checks, system level, line replacement unit (LRU) level and board level built-in-test (BIT).
- External monitors that form components such as external watchdog timers and external redundancy.
- Internal monitors form components that are placed within the devices such as BIT.

Using the concept of safety nets for COTS PEH and PDH (for which detailed design data is not available), it is important to examine the system architecture to identify the mechanisms that are used to capture PEH failures. These safety nets can be used within the PEH safety assessment to determine the type of PEH failures that are captured.

4. Petri nets: an overview

Petri nets have been widely used for modelling and analysing concurrent systems. They have both a graphical and mathematical form, hence, their models are amenable to formal analysis. Amongst the main attributes of this technique is its ability to explicitly capture the information flow, control flow or synchronisation within a system and analyse properties such as liveness (i.e. checking that a given state is reachable) and safety (i.e. absence of deadlock). Petri nets have been applied to numerous domains and they have also been successfully used to analyse software and hardware based systems [17,24].

The following provides a brief overview of Petri nets, a more detailed account is given in [14–16].

Definition 1. A marked Petri net N is defined as a 5-tuple: $N = \{P, T, I, O, M_o\}$ where:

$P = \{p_1, p_2, \ldots, p_n\}$, $n > 0$. The node p_n is known as a place.
$T = \{t_1, t_2, \ldots, t_m\}$ $m > 0$. The node t_m is known as transition.
I: $P \times T \rightarrow \{0, 1\}$. I is an input function that defines a set of directed arcs from P to T.
O: $T \times P \rightarrow \{0, 1\}$. O is an output function that defines a set of directed arcs from T to P.
M_o: $P \rightarrow \{0, 1, 2, \ldots\}$. M_o is an initial marking and represents the initial state of the net.

Petri net's graphical form is shown in Fig. 1 where a circle denotes a place, a bar denotes a transition, a black dot denotes a token and a single pointed arrow denotes an arc. The marking of N represents its state and this changes when a transition becomes