XMASCRAFT: a tool for GALS Modelling and Verification of Communication Circuits

Frank Burns*, Danil Sokolov† and Alex Yakovlev† Email:@newcastle.ac.uk
School of *Computing Science and †Electrical and Electronic Engineering, Newcastle University, UK.

The focus of XMASCRAFT is to provide a novel structured visual approach to GALS modelling and verification by providing a platform which integrates existing [1] and advanced GALS techniques into a unified environment for the analysis and visualisation of complex communication systems that are modelled using xMAS models [2], [3]. The intention of such an environment is to make it easier to gain visual insight into the causality of complex structural problems that may arise, such as deadlocks. The modelling and verification flow is shown in Fig. 1.
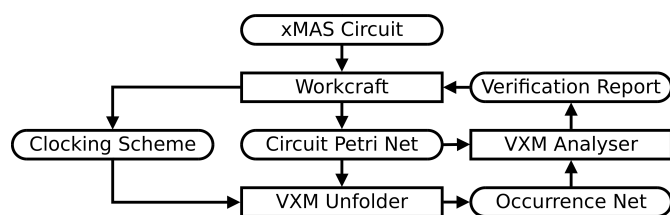

Fig. 1: XMASCRAFT tool flow.

Modelling is undertaken via a tool called WORKCRAFT [4]. Workcraft provides integration of a variety of plugins such as XMASCRAFT that aid in the visualisation/simulation of different graphical interpretations e.g. xMAS, Circuit Petri-nets [5], but which are linked. Therefore, the analysis of lower level models can be related back to the original model graphically. Inside the XMASCRAFT plugin GALS implementations are modelled using a specialised synchroniser primitive which allows for alternative synchronisation solutions. xMAS models are then translated to Circuit Petri-nets and a novel unfolding algorithm is deployed using a tool (VXM Unfolder) from Circuit Petri-nets to Structured Occurrence Nets (SONS) [6]. Deadlocks can be difficult to detect if significant parts of a GALS system become disabled. Structured Occurrence Nets were designed to enable visualisation of such complex behaviours which are more difficult to analyse and they are amenable for the modelling of structural links between modules. This is useful particularly when one wants to investigate point-to-point effects or how far the occurrence and effect of a problem extends between different modules.

A multi-level analyser (VXM Analyser) is used for verification. For this a novel concept of blocking/idle deadlock relations is introduced which describes how deadlocks in different parts of the system relate to each other. This representation, which is derived from the SONS model, enables more detailed structural visualisation of the deadlocks and their causality throughout the GALS communication system.

The XMASCRAFT plugin operating in deadlock relational analysis mode is shown in Fig. 2.
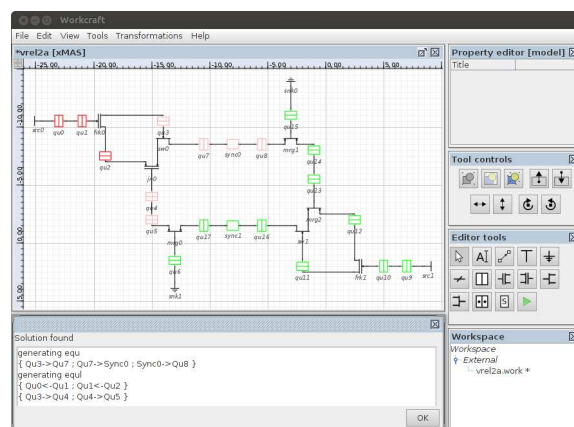

Fig. 2: XMASCRAFT

Via feedback through the XMASCRAFT interface, the novel relational representation enables structural analysis of deadlocks to be carried out across communication links to reveal various details including: vulnerable parts of the system which are susceptible to shutdown (i.e. partial/global deadlock); point-to-point causes of deadlock from one local module to another (using querying); multiple original causes of deadlocks and their visualisation in a single instance or snapshot and difficult to detect deadlocks that are hidden or masked by other deadlocks.

It can be clearly demonstrated via XMASCRAFT, through the use of querying, how the sources of the deadlocks can be resolved by changing specific components in the xMAS model thereby rendering the communication circuits deadlock free.

REFERENCES

[1] F. Burns, D. Sokolov and A. Yakovlev, "GALS Synthesis and Verification for xMAS models," *Proceedings of DATE'2015*, pp. –, 2015.

[2] S. Chatterjee, M. Kishinevsky and U. Ogras, "xMAS: quick formal modelling of communication fabrics to enable verification," *IEEE Design and Test of Computers*, Vol 29, No. 3, pp. 80–88, 2012.

[3] S. Joosten and J. Schmaltz, "Generation of inductive invariants from register transfer level designs of communication fabrics," *Proc. Formal Methods and Models for Codesign (MEMOCODE) 2013*, pp. 57–64, 2013.

[4] WORKCAFT homepage. *http://workcraft.org/*

[5] A. Yakovlev, L. Gomes and L. Lavagno: "Hardware design and Petri nets," *Springer*, 2000.

[6] M. Koutny and B. Randell, "Structured occurrence nets: a formalism for aiding system failure prevention and analysis techniques," *Proc. ACM Fundamenta Informaticae*, pp. 41–91, 2009.