XBM2PLA: A Flexible Synthesis Tool for Extended Burst Mode Machines

Oliver Kraus Martin Padeffke
Institute for Computer Aided Circuit Design
Friedrich-Alexander-University of Erlangen-Nuremberg

Abstract

*This paper describes the results of a new synthesis tool (*XBM2PLA*) for asynchronous state machines [2].* XBM2PLA *generates the boolean functions for an asynchronous circuit.* XBM2PLA *is compatible to several other tools but often generates smaller results. Moreover,* XBM2PLA *allows a more flexible description of asynchronous circuits than other existing tools.*

1. Introduction

The synthesis procedure for asynchronous state machines applies several restrictions on the specification of asynchronous circuits. R. Fuhrer and S. Nowick [4] defined a file format („burst mode specification“) which considers such restrictions and allows the specification of valid asynchronous state machines. An extension for this file format („extended burst mode specification“) has been introduced by Yun [10, 11].

At the moment there are two important programs that derive the boolean function of an asynchronous circuit from a burst mode specification: 3D and MINIMALIST. In the remaining part of the paper we will compare XBM2PLA with 3D and MINIMALIST according to the input format and the results.

2. Input Format

The „burst mode specification“ is a simple file format which describes the behaviour of a state machine and its output signals. Transitions of the state machine are initiated by falling or rising edges of input variables. Rising and falling edges are written as v^+ and v^- (Figure 1). There are also some more restrictions which can be studied in [4]. This basic file format can be processed by 3D, MINIMALIST and XBM2PLA.

The „extended burst mode specification“ additionaly allows „conditionals“ and „directed don't cares“.

- Conditionals: A variable can be checked for a logical value. This is written as $[v^+]$ for logical high and $[v^-]$ for logical low.
- Directed don't cares: A variable might either keep its state or might change exactly once. This is written as $v*$.

This file format can be parsed by 3D if the „extended burst mode specification“ meets another condition: A variable must either use $[v^+]$ ($[v^-]$) or v^+ (v^-). The mixture of $[v^+]$ ($[v^-]$) with v^+ (v^-) is not allowed.

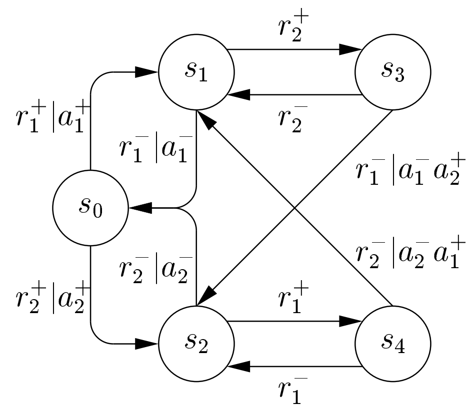

Figure 1. Burst-mode specification (Mutual exclusion)

For XBM2PLA, we removed this additional restriction. Any mixture is allowed: Conditionals ($[v^+]$, $[v^-]$) and edge transitions (v^+, v^-). Figure 2 is an example for such a mixture. It shows an equivalent specification of Figure 1.

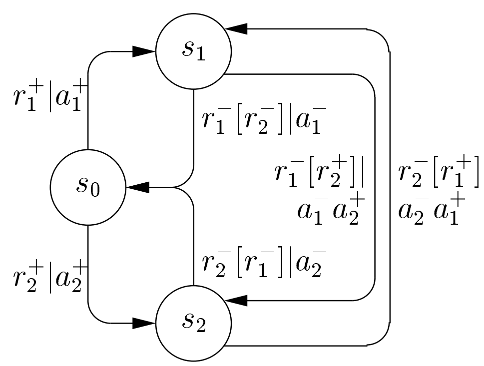

Figure 2. Extended burst mode specification (Mutual exclusion)

Both specifications can be reduced by XBM2PLA to the same boolean functions:

$$\begin{aligned} a_1 &= r_1 \cdot \overline{a_2} + r_1 \cdot \overline{r_2} = r_1 \cdot (\overline{a_2} + \overline{r_2}) \\ a_2 &= r_2 \cdot a_2 + \overline{r_1} \cdot r_2 = r_2 \cdot (a_2 + \overline{r_1}) \end{aligned}$$

Further simplification will lead to the circut in figure 3.

3. Optimization Procedure

In general, XBM2PLA tries to find an minimal solution for all subproblems. XBM2PLA applies the following steps:

1. Construction of the state machine [3].

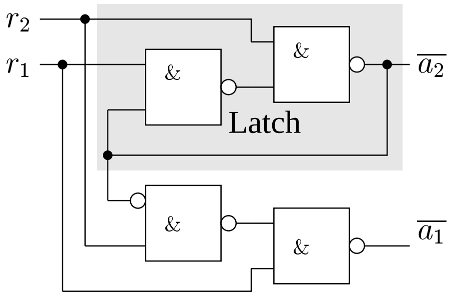

Figure 3. Digital circuit (Mutual exclusion)

2. State-minimization [5].
3. Minimal asynchronous state encoding [9, 7]. The output values of the state machine are used for dichotomy reduction and state separation [4].
4. Exact and hazardfree minimzation of the boolean machine function [6].

XBM2PLA includes a new approach for the asynchronous state encoding: If a cube is interpreted as a dichotomy, the following function calculates every „prime“ (largest) dichotomy from an initial list. If $\mathbf{F}$ is binate:

$$\text{PD}(\mathbf{F}) = \text{SCC}(x_i \cdot \text{PD}(\mathbf{F}_{x_i}) \cup \overline{x_i} \cdot \text{PD}(\mathbf{F}_{\overline{x_i}})).$$

If $\mathbf{F}$ is unate: $\text{PD}(\mathbf{F}) = \bigcap_{B_i \in \mathbf{F}} B_i$. A cover algorithm selects a minimum subset of these dichotomies.

4. Results

The following tables use two key values for comparison: The number of additional state bits (sb) and the sum of literals and product terms (l+p). Table 1 shows the results of the synthesis procedure for several extended burst mode descriptions.

		3D		XBM2PLA	
	in/st/out	sb	l+p	sb	l+p
alu1	3/7/5	2	63	2	74
alu2	5/14/7	3	187	2	138
mul1	3/4/3	1	61	2	47
mul2	3/3/3	0	25	0	19
biu-dma2fifo	4/7/2	3	69	4	79
fifocellctrl	2/3/2	1	16	1	16
scsi-init-send	5/9/3	3	109	3	96
dff	2/4/1	2	19	1	30
sbuf-ram-write	5/6/5	1	47	1	51
ram-read-sbuf	5/7/5	0	33	0	32
sbuf-send-pkt2	3/4/2	2	27	1	24
		18	656	17	606

Table 1. Examples: Extended Burst Mode

Table 2 lists the results of the synthesis procedure for some larger non-extended burst mode machines.

3D contains very good heuristic procedures. Often 3D generates a result which is nearly as good as an exact calculation. 3D also allows the calculation of larger circuits where MINIMALIST produces an out-of-memory error (freq8). MINIMALIST often finds the minimal solution by increasing the number of state bits. In contrast XBM2PLA always uses the calculated minimum number of state bits which sometimes creates a larger circuit (ring-counter).

	3D		MINIMALIST		XBM2PLA	
	sb	l+p	sb	l+p	sb	l+p
dram-ctrl	1	71	0	51	0	62
pscsi-isend	4	105	3	70	3	90
pscsi-tsend	4	77	3	66	3	76
scsi-tsend-bm	2	92	2	89	2	100
pe-send-ifc	2	89	3	87	3	100
ring-counter	1	220	2	137	1	217
bincnt3	2	58	9	152	3	55
freq7	9	112	4	72	4	74
freq8	11	135	error		4	89

Table 2. Examples: Burst Mode

5. Conclusion

XBM2PLA accepts the most flexible description and usually calculates smaller or similar results compared to 3D and MINIMALIST. A further optimization of XBM2PLA might include new and heuristic algorithms for state encoding [1] and hazardfree minimization [8]. XBM2PLA is a part of the DGC project [2] and is published under the GNU general public license.

References

[1] O. Coudert. A New Paradigm for Dichotomy-based Constrained Encoding. In *Proc. Design, Automation and Test in Europe (DATE)*, pages 830–834, 1998.

[2] `http://dgc.sourceforge.net`.

[3] R. Fuhrer, B. Lin, and S. Nowick. Symbolic hazardfree minimization and encoding of asynchronous finite state machines. In *Proc. International Conf. Computer-Aided Design (ICCAD)*. IEEE Computer Society Press, 1995.

[4] Fuhrer, Nowick. *Sequential Optimization of Asynchronous and Synchronous Finite-State Machines: Algorithms and Tools*. Kluwer Academic Publishers, 2001.

[5] C. J. Myers. *Asynchronous Circuit Design*. John Wiley & Sons, 2001.

[6] S. M. Nowick and D. L. Dill. Exact two-level minimization of hazard-free logic with multiple-input changes. *IEEE Transactions on Computer-Aided Design*, 14(8):986–997, Aug. 1995.

[7] A. Saldanha, T. Villa, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. A Framework for Satisfying Input and Output Encoding Constraints. In *Proc. ACM/IEEE Design Automation Conference*, pages 170–175. IEEE Computer Society Press, 1991.

[8] M. Theobald and S. M. Nowick. Fast heuristic and exact algorithms for two-level hazard-free logic minimization. *IEEE Transactions on Computer-Aided Design*, 17(11):1130–1147, 1998.

[9] J. H. Tracey. Internal state assignment for asynchronous sequential machines. *IEEE Transactions on electronic computers*, 15(4):551–560, Aug. 1966.

[10] K. Yun, D. Dill, and S. Nowick. Synthesis of 3D Asynchronous State Machines. In *Proc. International Conf. Computer Design (ICCD)*, pages 346–350, 1992.

[11] K. Y. Yun. Synthesis of asynchronous controllers for heterogeneous systems. Technical Report CSL-TR-94-644, Dept. of electrical engineering, Stanford University, 1994.