

时序模型提取中复杂单元拓扑结构识别的实现研究

段 芬^{1,2}, 金湘亮^{1,2}

(1 湘潭大学 物理与光电工程学院, 湖南 湘潭 411105; 2 湖南省微光电与系统集成工程实验室, 湖南 湘潭 411105)

摘 要: 基于 NanoTime 对不同复杂 D 触发器和多米诺机构采取不同的处理方式, 解决了单元拓扑结果识别错误的问题. 结合一个单元的实际案例进行仿真验证, 结果显示基于所述的处理方式可以提取出有效的单元时序模型. 目前相应的设计已投入流片, 成功应用于实际工程应用中, 这也说明其具有可靠的实际应用价值, 对工程应用设计具有重要意义.

关键词: 时序模型; NanoTime; 标记与擦出

中图分类号: TN4

文献标识码: A

文章编号: 1000—7180(2018)01—0086—05

The Methodology and Implementation for Recognition of Complex Topology During Timing Model Extraction

DUAN Fen^{1,2}, JIN Xiang-liang^{1,2}

(1 School of Physics and Optoelectronics, Xiangtan University, Xiangtan 411105, China;

2 Hunan Engineering Laboratory for Microelectronics, Optoelectronics and System on a Chip, Xiangtan 411105, China)

Abstract: In the process of chip design, timing model extraction is one of the key of static timing analysis. Different complex D flip-flop and domino structure need different approaches based on the NanoTime to solve complex topology recognition error, which improves the effectiveness of timing model. The simulation results are shown using an actual case. The simulation data shows that effective timing model can be achieved according to the method provided by this paper. At the present, the corresponding design flow has been invested and successfully applied in practical engineering implementation. This also means it has practical applications value of reliability and it is necessary for engineering implementation.

Key words: timing model; NanoTime; mark and erase

1 引言

在芯片设计过程中, 静态时序分析通常是必不可少的一环, 而静态时序分析的基础就是单元时序库信息, 时序模型的准确性对于芯片设计至关重要, 是整个设计流程能否顺利进行的关键因素之一^[1-2]. 对于各种结构复杂的单元, 比如多输入多时钟触发器、多米诺等单元, 它们的时序模型提取就是一个难点. 通用时序模型提取工具主要有 Siliconsmart 和 NanoTime(NT) 两种. Siliconsmart 是用真值表或者函数的形式去描述某个电路的特征来获取相应结

构的时序模型, 对于复杂单元, 即使通过很复杂的方式描述了电路特征, 但是提取出来的时序信息依然存在缺失或者错误的情况^[3]. 因此对于复杂单元, 通常采用含有识别复杂结构机制的 NT 去提取复杂单元的时序模型. NT 通过自动生成符合 liberty 语法规范格式的 timing library 文件, IP 设计用户可以根据应用的实际情况设置输入端信号的转换时间和输出端口的负载, 根据设置值在 lib 文件中查表即可获取该模块内部准确的时序信息. NT 定位于全定制数字逻辑电路的静态时序分析和 timing library 生成, 因此它拥有强大的逻辑电路拓扑识别能力,

收稿日期: 2017—03—17; 修回日期: 2017—04—22

基金项目: 国家自然科学基金项目(61233010); 湖南省自然科学基金资助项目(2015JJ1014)

可自动识别传输门,反馈管,上/下拉管,交叉耦合上拉管,弱上拉管,开关结构,锁存器,触发器,多路选择器,多米诺预充,随机存储器,寄存器等大多数常见的电路结构^[4].但是对于一些结构复杂的单元,在对它们自动识别时,存在一定的错误,不同的特殊结构需要对其进行不同的特殊处理.本文就是根据不同的复杂 D 触发器(DFF)结构和多米诺预充结构,采取不同的标记与擦除方式,从而准确地提取出时序模型.

2 特殊结构的辅助识别方法

2.1 NT 提取时序模型的流程

NT 提取时序模型的流程如图 1 所示.其主要工作有:(1) NT 环境的配置,包括了网表、环境参数、端口、仿真参数等设置;(2) 单元拓扑结构识别;(3) 路径报告分析,验证所提取模型的正确性;(4) 验证无误时,获取最终的时序模型^[5-6].

在整个时序模型的提取过程中,NT 必须能够正确识别出整个电路里所含有的结构,通常这部分工作占用了 80%的时间. NT 只有正确识别了电路所包含的结构,才能对其进行正确的时序约束并分析,进而提取出准确度高的时序模型.因此结构识别的正确性决定了最终时序模型的正确性^[7-8].

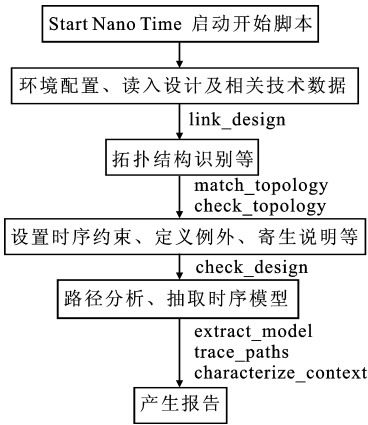


图 1 时序模型的提取流程

2.2 复杂 DFF 结构的处理

2.2.1 错误识别结构的删除

提取时序模型过程中,复杂 DFF 单元的许多结构会被错误识别,主要是被错误识别成如下两种情况.

第一种是图 2 所示的单元,这类由两个传输门(图 2 中传输门 T1 和 T2)输出端接在一起的拓扑结构会被 NT 自动识别为 mux 结构.从这个电路结构上分析,此种结构通常是 latch 的时钟控制门,因

此需要擦除被错误识别的 mux 结构.

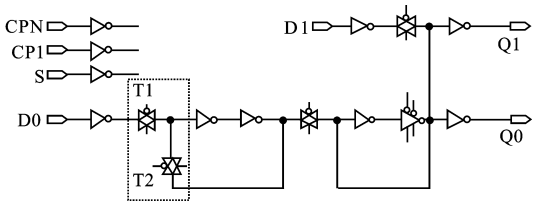


图 2 复杂 DFF 结构中两传输门的输出相接结构

第二种是图 3 所示的单元,两个带钟控反相器(图 3 中反相器 IC1 和 IC2)的输出端直接相连,这种拓扑结构会被 NT 自动识别为 turn off 结构.但是从结构上分析,它通常是 latch 的一个主从锁存时钟控制,因此需要对已识别出的 turn off 结构进行擦除.

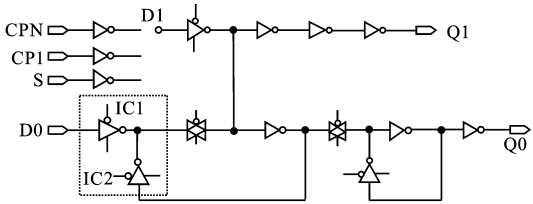


图 3 复杂 DFF 结构中两个带钟控的反相器直接相接结构

2.2.2 标记未识别的 latch 或 flip-flop 结构

提取时序模型过程中,有的单元由于某些特殊结构,NT 无法正确识别,都需要手动标记出 latch 或者 flip-flop.

如图 3 所示的单元,对错误识别的 turn off 结构进行擦除之后,需要手动标记主锁存结构.接着经过 match_topology 命令,NT 才能正确识别出整个 flip-flop 结构,提取出正确的时序模型.

对于图 4 所示的单元,它的锁存结构(如图虚线框)输出位于反相器 I2 之前,NT 能正确识别出前后两个锁存结构,却不能识别出整个 flip-flop 结构,需要手动标示出整个 flip-flop 结构,才能提取出此单元的完整时序模型.

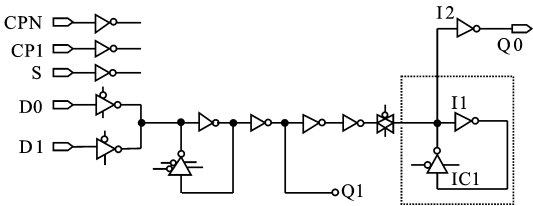


图 4 复杂 DFF 结构中锁存输出位于反相器前的结构

2.2.3 复杂 D 触发器、Latch 与门控复用结构的分步处理

有的单元中,存在 D 触发器、Latch 与门控复用

情况。D 触发器的主锁存结构只有在扫描链打开情况下才能进行工作,所以在扫描链打开与关闭的情况下,需要不同的时序弧。在提取此类单元时序模型时,对其进行分情况提取,最后再进行合成。

如图 5 所示的结构,它的门控 RI1 结构与 D 触发器结构不能同时存在,将其分两步进行时序模型的提取:扫描链时钟 CPN 处于工作状态时,门控时钟 C 为 0,仅有一个 D 触发器的功能。提取时序模型时,只需要标记出前后 latch 以及 D 触发器结构;扫描链 CPN 处于非工作状态时,电路只有门控 RI1 与从锁存结构(图中大虚线框)工作,此时将时钟 CPN 设置为 0,标记出门控结构以及从锁存结构,同时设置数据输入穿过 latch 到输出为伪路径,得到门控的建立、保持时间以及 CLK→OUT 的时序信息。将两者合成,形成此单元的完整时序模型。

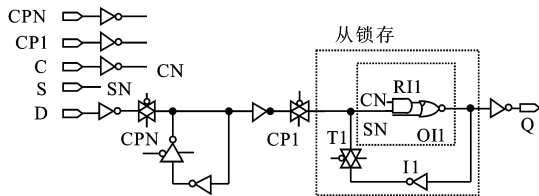


图 5 复杂 D 触发器、Latch 与门控复用结构

有的单元由于自身结构的特殊性,在原结构条件下无法建立出某些需要的时序信息,如图 6 所示的单元。提取 D0 对时钟 CPN 的建立保持时间时,常规处理方式将会产生错误。对此结构进行分析,建立 D0 对时钟 CPN 的时序弧时,单元中的其他结构可视为整体负载,因此可将反向器 I1 与放到传输门 T1 之前,如此处理后可建立出 D0 对时钟 CPN 的时序弧,最后再将这个时序信息合并到原时序模型中,得到此单元的完整时序模型。

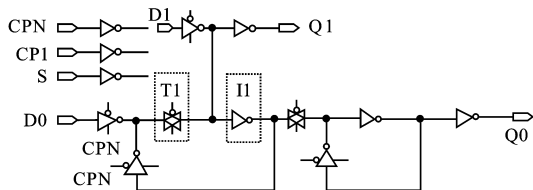


图 6 复杂 DFF 结构中反相器与传输门置换结构

2.2.4 多时钟单元识别

一些单元存在三个时钟端口,如图 7 所示的单元。主从锁存各自含有控制传输门的一个时钟(图 7 中信号 1 与 2),还有一个时钟信号在主锁存内部(图 7 中信号 3),与其他三路组成了一个多路选择器,且其他三个多路选择器的控制信号是由时钟 S0

生成。时钟 S0 分别进入与非门 AI1、AI2、AI3 产生三个控制信号,当时钟 S0 为高时,D0 通路打开,与此同时与非门 AI1、AI2、AI3 产生的另外三个控制信号恒为 0,通路 D1、D2、D3 关闭;当时钟 S0 为低,则通路 D1、D2、D3 的控制信号取决于各自的输入。对此类 mux 进行四个控制信号的 one hot 设置,使得 NT 可以正确识别出此单元结构,从而正确提取出此单元时序模型。

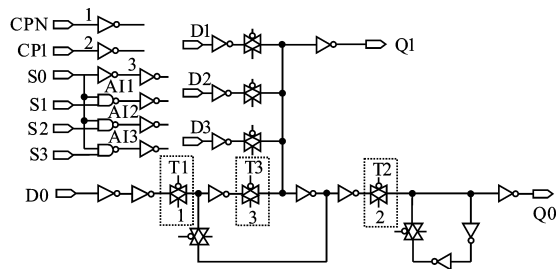


图 7 多时钟的复杂 DFF 结构

2.3 多米诺结构的处理

2.3.1 预充结构输出进行逻辑运算的标识方式

对于某些多米诺结构,提取其时序模型时,内部一些晶体管需要标记方向。此时对晶体管信号流向进行分析,然后对其进行标记。

某些多米诺单元中,它的预充(precharge)结构输出不是直接作为端口输出,而是进入逻辑门与其他信号进行逻辑运算,此时 NT 无法识别这个 precharge 结构。例如图 8 所示单元的 precharge 结构 P1,它需要手动标记: mark_precharge = type NMOS - latch_type half_latch - evaluate_net xx - precharge_clock xx - keeper_data xx - evaluate_data xx - evaluate_clock xx,从而完成此单元的 precharge 结构识别。

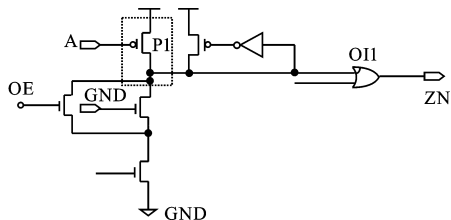
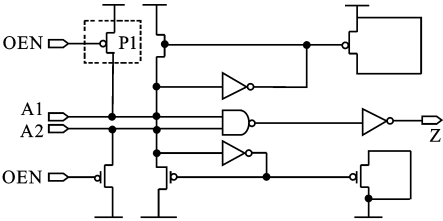


图 8 多米诺结构示意图

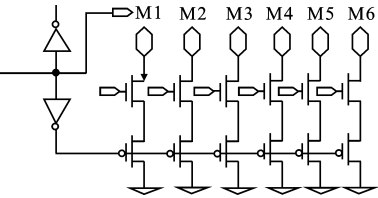
2.3.2 标记大量多米诺结构的方式

在图 9 所示的单元中,多米诺结构的 precharge 电路(如图 9(a) 所示)和运算电路(如图 9(b) 所示)分别处在不同单元中,且每一个 precharge 电路都连接着多个不同的运算电路,所连接的运算单元的实例号和单元名都不相同时,NT 无法直接正确标

记出相应的拓扑结构. 此种情况下,单独标记其中一个 precharge 结构,接着使用 `-match_pattern` 选项,最后对所有类似的结构进行搜索并标记: `mark_precharge -type NMOS -latch_type half_latch -evaluate_net xx -precharge_clock xx -keeper_data xx -evaluate_data xx -match_pattern`, 由此可以实现整个 precharge 结构的批量识别.



(a) 预充部分



(b) 运算部分

图 9 多米诺结构中的预充与运算部分

2.3.3 钟控门的标记与擦除

在有的单元结构中,存在钟控门不被 NT 识别的情况. 此时,首先需要判断是否正确设置时钟信号端口,如果时钟设置无误,钟控门仍然没有被自动识别,则需要手动识别. 如图 10 所示的单元, SON 是一个时钟信号, S1 是输入端口信号. 如果对此种结构进行常规处理,端口 S1 的时序库为空. 因为 NT 无法自动识别出 OI1 和 OI2 为钟控门,数据 S1 相对时钟 SON 进行时序检查时,NT 对其没有进行分析. 这种情况下,需要手动标记与非门 OI1、OI2 为钟控门,使 NT 能正确识别出此门控结构.

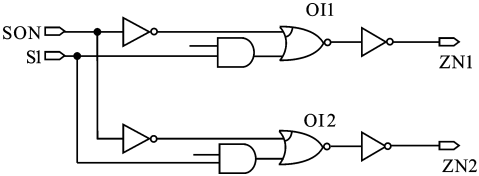


图 10 钟控门建立时间检查的电路

3 仿真验证结果

以图 6 所示的单元为例,根据此单元结构的特殊性,结合 2.2.3 节所述的解决方法:将图 6 中反向

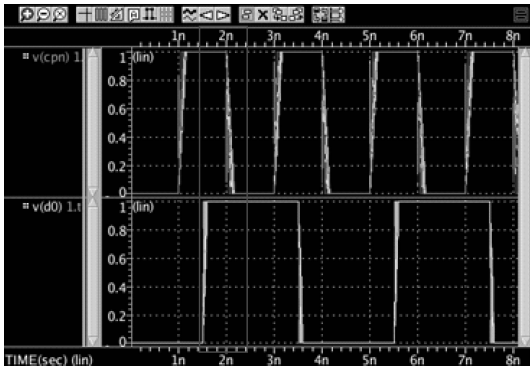
器 I1 与放到传输门 T1 之前,建立出 D0 对时钟的时序弧. 利用 NT 对此单元进行时序信息的提取,采用 FF 的 MOS 管模型,温度为 0℃,电压为 1.1 V, 以此建立出 D0 对时钟 CPN 的延时数据的查找表如图 11 所示.

```
timing_type: setup_falling;
rise_constraint(f_dtrans_ctrans){.
  index_1({0.002600, 0.010600, 0.024600, 0.048800, 0.097600, 0.216000, 0.559600});.
  index_2({0.002600, 0.010600, 0.024600, 0.048800, 0.097600, 0.216000, 0.559600});.
  values({0.079984, 0.079960, 0.079930, 0.079882, 0.079809, 0.079642, 0.079141});.
  "0.083192, 0.083169, 0.083139, 0.083091, 0.083018, 0.082851, 0.082350";.
  "0.087508, 0.087485, 0.087453, 0.087406, 0.087333, 0.087166, 0.086665";.
  "0.091759, 0.091736, 0.091706, 0.091658, 0.091585, 0.091418, 0.090917";.
  "0.095175, 0.095152, 0.095122, 0.095074, 0.095001, 0.094834, 0.094333";.
  "0.091344, 0.091321, 0.091291, 0.091243, 0.091170, 0.091003, 0.090502";.
  "0.054777, 0.054754, 0.054724, 0.054676, 0.054603, 0.054436, 0.053935";.
fall_constraint(f_dtrans_ctrans){.
  index_1({0.002600, 0.010600, 0.024600, 0.048800, 0.097600, 0.216000, 0.559600});.
  index_2({0.002600, 0.010600, 0.024600, 0.048800, 0.097600, 0.216000, 0.559600});.
  values({0.099141, 0.099118, 0.099088, 0.099040, 0.098967, 0.098799, 0.098299});.
  "0.103151, 0.103128, 0.103098, 0.103050, 0.102977, 0.102810, 0.102309";.
  "0.108272, 0.108249, 0.108219, 0.108171, 0.108098, 0.107931, 0.107430";.
  "0.119294, 0.119271, 0.119241, 0.119193, 0.119120, 0.118953, 0.118452";.
  "0.139433, 0.139410, 0.139380, 0.139332, 0.139259, 0.139092, 0.138591";.
  "0.182016, 0.181993, 0.181963, 0.181914, 0.181841, 0.181674, 0.181174";.
  "0.284743, 0.284720, 0.284690, 0.284642, 0.284569, 0.284402, 0.283901";.
timing_type: hold_falling;
rise_constraint(f_dtrans_ctrans){.
  index_1({0.002600, 0.010600, 0.024600, 0.048800, 0.097600, 0.216000, 0.559600});.
  index_2({0.002600, 0.010600, 0.024600, 0.048800, 0.097600, 0.216000, 0.559600});.
  values({-0.014737, -0.014719, -0.014690, -0.014650, -0.014598, -0.014443, -0.013793});.
  "-0.017647, -0.017629, -0.017600, -0.017559, -0.017507, -0.017353, -0.016703";.
  "-0.021007, -0.020988, -0.020960, -0.020919, -0.020867, -0.020713, -0.020063";.
  "-0.023190, -0.023171, -0.023143, -0.023102, -0.023050, -0.022896, -0.022246";.
  "-0.022789, -0.022770, -0.022742, -0.022701, -0.022649, -0.022495, -0.021845";.
  "-0.011226, -0.011208, -0.011179, -0.011139, -0.011087, -0.010932, -0.010282";.
  "-0.041923, 0.041914, 0.041970, 0.042011, 0.042063, 0.042217, 0.042867";.
fall_constraint(f_dtrans_ctrans){.
  index_1({0.002600, 0.010600, 0.024600, 0.048800, 0.097600, 0.216000, 0.559600});.
  index_2({0.002600, 0.010600, 0.024600, 0.048800, 0.097600, 0.216000, 0.559600});.
  values({-0.027562, -0.023169, -0.017759, -0.010737, 0.000957, 0.025046, 0.089391});.
  "-0.031185, -0.026793, -0.021382, -0.014361, -0.002667, 0.021423, 0.085767";.
  "-0.036089, -0.031696, -0.026286, -0.019264, -0.007570, 0.016520, 0.080864";.
  "-0.046524, -0.042131, -0.036721, -0.029700, -0.018005, 0.006084, 0.070429";.
  "-0.064090, -0.059698, -0.054287, -0.047266, -0.035572, -0.011482, 0.052863";.
  "-0.101055, -0.096662, -0.091252, -0.084231, -0.072536, -0.048447, 0.015898";.
  "-0.192990, -0.188597, -0.183187, -0.176166, -0.164471, -0.140382, -0.076037";.
}
```

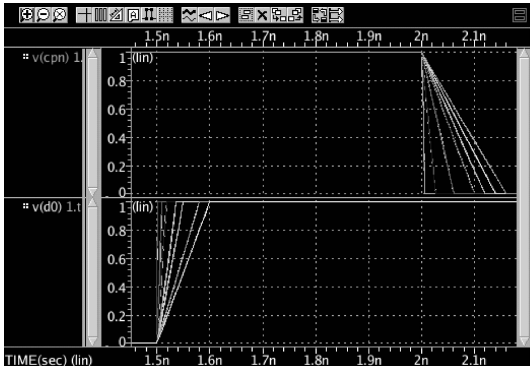
图 11 NT 建立的 D0 对 CPN 的延时查找表

为验证其精准性,采用 HSPICE 仿真工具对 D0 到 CPN 的建立时间以及保持时间 `rise_constraint`, `fall_constraint` 进行测量,其配置环境与 NT 所配置的环境一致. 在测量结果中,以建立时间的 `rise_constraint` 为例,表 1 为 NT 所获取的时序延时 `setup_rise` 查找表,图 12(a)为 HSPICE 仿真结果,图 12(b)为(a)中白色框的扩大图,7 条不同的波形线分别表示不同的转换时间,以此测量出相应的 D0 对 CPN 的时序延时,具体如表 2 所示,表 3 为 NT 与 Hspice 获取的延时误差表.

由表 3 可知,NT 所得到的 `setup/hold` 值, D0 对 CPN 的延时与 Hspice 仿真结果的时序逻辑单元误差在 4.6 ps 以内,在工程应用的允许范围内,由



(a)



(b)

图 12 HSPICE 仿真结果

此证明本文根据单元特点所提出的相关处理方式的有效性.

表 1 NT 所获取的时序延时 setup_rise 查找表							ns
0.079 984	0.079 96	0.079 930	0.079 882	0.079 809	0.079 642	0.079 141	
0.083 192	0.083 169	0.083 139	0.083 091	0.083 018	0.082 851	0.082 35	
0.087 508	0.087 485	0.087 455	0.087 406	0.087 333	0.087 166	0.086 665	
0.091 759	0.091 736	0.091 706	0.091 658	0.091 585	0.091 418	0.090 917	
0.095 175	0.095 152	0.095 122	0.095 074	0.095 001	0.094 834	0.094 333	
0.091 344	0.091 321	0.091 291	0.091 243	0.091 17	0.091 003	0.090 502	
0.054 777	0.054 754	0.054 724	0.054 676	0.054 603	0.054 436	0.053 935	

表 2 Hspice 所获取的时序延时 setup_rise 查找表							ns
0.081 532	0.081 664	0.161 594	0.081 797	0.081 919	0.161 561	0.081 958	
0.084 75	0.084 907	0.168 046	0.085 084	0.085 528	0.168 379	0.085 65	
0.089 178	0.089 336	0.176 791	0.089 516	0.089 623	0.176 789	0.090 065	
0.093 504	0.093 647	0.185 353	0.093 818	0.093 935	0.185 353	0.094 647	
0.097 305	0.097 442	0.192 564	0.097 624	0.097 731	0.192 565	0.098 313	
0.094 154	0.094 241	0.185 532	0.094 413	0.094 53	0.185 533	0.094 712	
0.058 037	0.058 124	0.112 848	0.058 216	0.058 423	0.112 859	0.058 455	

表 3 NT 与 Hspice 获取的延时误差表							ns
0.001 548	0.001 704	0.001 883	0.001 915	0.002 11	0.002 46	0.002 817	
0.001 558	0.001 738	0.001 906	0.001 993	0.002 51	0.002 43	0.003 3	
0.001 67	0.001 851	0.002 018	0.002 11	0.002 29	0.002 61	0.003 4	
0.001 745	0.001 911	0.002 08	0.002 16	0.002 35	0.002 7	0.003 73	
0.002 13	0.002 29	0.002 49	0.002 55	0.002 73	0.003 1	0.003 98	
0.002 81	0.002 92	0.003 11	0.003 17	0.003 36	0.003 79	0.004 21	
0.003 26	0.003 37	0.003 58	0.003 54	0.003 82	0.004 14	0.004 52	

4 结束语

本文着重讲述在利用 NanoTime 进行时序模型提取时,解决对复杂 DFF 结构和多米诺预充结构识别错误的问题,从而正确提取出相应单元的完整时序模型.最后以图 6 所示的复杂 DFF 结构为例,实际建立出 D0 对时钟 CPN 的时序弧,得出此单元拓扑结构的时序模型.此外通过 HSPICE 仿真工具对 D0 到 CPN 的建立时间以及保持时间进行测量,以验证所述处理方法的有效性.

结合工程经验,复杂 D 触发器结构和多米诺结构基本是本文所示结构的衍生.因此通过本文所述方法,结合实际情况运用,基本可以解决复杂拓扑结构识别错误的问题.

参考文献:

[1] Jouppi N P . Timing analysis and performance improvement of MOS VLSI designs[J]. IEEE Trans, Computer-Aided Design Integrated Circuits and System, 1987, 6(4),650-665.

[2] BlaauwD , Chopra K , Srivastava A, et al, Statistical timing analysis; from basic principles to state of the art[J]. IEEE Trans, Computer- Aided Design Integrated Circuits and System, 2008, 27(4):589-607.

[3] 袁甲,张苏敏,商新超,等. 一种基于商用标准单元库的极低电压电路设计方法[J]. 微电子学与计算机, 2014, 31(12):56-59.

[4] 刘峰. 集成电路静态时序分析与建模[M]. 北京:机械工业出版社,2016.

[5] 王东,陈岚,柳臻朝,等. 基于 40nm CMOS 工艺的 DAC IP 核物理与时序建模[J]. 微电子学与计算机, 2015,32(2):56-59.

[6] 王东,陈岚,冯燕. 一种基于 40nm CMOS 工艺的电流舵 DAC IP[J]. 微电子学与计算机, 2017, 34(2): 25-29.

[7] 陈超文. 面向光纤通道的 SerDes 电路 IP 化技术研究[D]. 成都:电子科技大学,2014.

5 结束语

本文使用使用基于 Fisher 准则和特征聚类的方法进行特征选择,使用 OS-KELM 实现对人的六种姿态分类识别. 特征选择大大降低了特征集的维度,降低了学习难度. OS-KELM 在 OS-ELM 的基础上引入核函数,补偿 OS-ELM 算法因隐层输入随机矩阵带来的误差. 随后对这两个算法从识别精度和训练、测试时间上进行了比较. 实验数据来自智能手机中的加速计和陀螺仪传感器. 实验结果显示 OS-KELM 的识别精度在 64 个特征/200 个神经元的条件下可达到 91.89%,相较已有成果的 82%和选取 Sigmoidal 函数为激活函数的 OS-ELM 算法的 88.00%,在精度上都有所提高. 同时,由于核函数的引入在训练时间和测试时间上也略有增加.

本文使用 OS-KELM 算法对人的姿态分类识别进行了探索,考虑进一步的研究方向如下:(1)连贯姿态的识别,很多动作是连贯在一起的,例如走路后上楼,然后下楼再站立等,这些类似情况会导致两个或多个姿态的交叉融合;(2)可以更进一步探索不同的核函数对识别精度和训练时间的影响.

参考文献:

- [1] Perez A J, Labrador M A, Barbeau S J. G—sense: a scalable architecture for global sensing and monitoring [J]. IEEE Network, 2010, 24 (4): 57-64.
- [2] Shi X, Xiong Q Y, Lei L N. Fall detection system based on pressure sensor[J]. Chinese Journal of Scientific Instrument, 2010, 31(3): 715-720.
- [3] Liu Z G, Song Y N, Shang Y. Posture recognition algorithm for theelderly based on BP neural networks[C]

// Proceedings of 27th Control and Decision Conference (CCDC). IEEE, 2015: 1447-1449.

- [4] Al Jeroudi Y, Ali M A, Latief M, et al. Online sequential extreme learning machine algorithm based human activity recognition using inertial data[C]// Control Conference (ASCC), 2015 10th Asian. IEEE, 2015: 1-6.
- [5] Deng W Y, Ong Y S, Tan P S, et al. Online sequential reduced kernel extreme learning machine[J]. Neurocomputing, 2016, 174: 72-84.
- [6] Anguita D, Ghio A, Oneto L, et al. A public domain dataset for human activity recognition using smartphones[C]//ESANN. Bruges, 2013.
- [7] 王颀, 郑链. 基于 Fisher 准则和特征聚类的特征选择[J]. 计算机应用, 2007, 27(11): 2812-2813.
- [8] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: theory and applications[J]. Neurocomputing, 2006, 70(1): 489-501.
- [9] Huang G B, Zhou H, Ding X, et al. Extreme learning machine for regression and multiclass classification[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2012, 42(2): 513-529.

作者简介:

韩 莹 女, (1991-), 硕士研究生. 研究方向为数字信号处理、无线通信.

张 浩(通讯作者) 男, (1975-), 副研究员. 研究方向为复杂环境下的线性和非线性信号处理技术.

E-mail: haozhang@ime. ac. cn.

刘 建 男, (1977-), 副教授. 研究方向为认知无线网络、移动网网络、物联网技术.

陈立平 男, (1983-), 助理研究员. 研究方向为信号处理、无线通信.

(上接第 90 页)

- [8] 孟祥志, 杨萌, 来金梅. 基于累积频数统计的 FPGA 互连资源时序库[J]. 计算机工程, 2013, 39(2): 254-259.

作者简介:

段 芬 女, (1993-), 硕士. 研究方向为静态时序分析.

E-mail: 18216405639@163. com.

金湘亮 男, (1974-), 博士, 教授. 研究方向为 SOC 设计、系统设计.