



ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

آزمایش یک سیستم روی یک تراشه با میکروپروسور تعبیه شده

چکیده

در این مقاله، ما روش آزمون برای هسته های تعبیه شده بر اساس سیستم روی یک تراشه (SoC) را که شامل یک هسته میکروپروسور می شود آزمایش می کنیم. ابتدا هسته میکروپروسور برای صحت تمام دستورالعمل ها آزمایش می شود و سپس قدرت محاسبه هسته میکروپروسور برای آزمایش حافظه های روی تراشه و دیگر هسته ها استفاده می شود. یک مجموعه آزمون iddq نیز برای آشکارسازی عیوب فیزیکی استفاده می شود و ویژگی های طراحی برای تسهیل آزمون Iddq توصیف می شوند.

۱. مقدمه

در سال های اخیر، فناوری ASIC از فلسفه تنظیم-تراشه به یک مفهوم سیستم روی یک تراشه مبتنی بر هسته های تعبیه شده (SoC) تکامل یافته است. یک SoC شامل بلوک های وظیفه ای قابل استفاده مجدد مختلف به نام هسته هایی مانند حافظه ها، میکروپروسورها، DSPها، کنترل باس و واسطه هایی مانند PCI و USB می شود. در حالیکه استفاده از هسته ها در SoC به عنوان گستره وسیعی از کاربردها به کار گرفته می شود، پیچیدگی این تراشه ها بسیار فراتر از آنست که توسط روش های مرسوم آزمایش شود [1]. مرجع [2]، یک مرور کلی مناسب را برای مشکلات آزمون فراهم می کند. یک مقدار اساسی برای جستجو، برای پرداختن به مشکلات آزمون، مورد بررسی است [1-5]. در تولید SoC، ترکیبی از روش های آزمون مانند آزمون وظیفه ای، اسکن کامل، BIST، Iddq و غیره استفاده می شوند. به معنای گسترده تر، هسته های فردی توسط یک یا چند روش زیر آزمایش می شوند:

۱. آزمایش یک هسته از طریق آزمایش وظیفه ای سیستم-تراشه

۲. کاربرد آزمون مستقیم در حین دسترسی به هسته از طریق مالتی پلکس نمودن پین I/O

۳. کاربرد آزمون برای هسته از طریق اسکن مرز موضعی یا یک ثبات؛

۴. اسکن نمودن خودآزمون توکار از طریق انواع مکانیزم های دسترسی و کنترل

۵. راه حل های مناسب؛

در این مقاله، ما یک روش آزمون را برای SOC توصیف می کنیم که شامل یک هسته میکروپروسور می شود. روش آزمون می تواند به چهار زیرمجموعه تقسیم شود: (i) آزمایش هسته میکروپروسور توسط اطمینان از صحت تمام دستورالعمل های آن؛ (ii) استفاده از هسته میکروپروسور برای آزمایش حافظه های تعبیه شده؛ (iii) استفاده از هسته میکروپروسور برای آزمایش دیگر هسته ها؛ (iv) پیاده سازی و طراحی برای آزمایش Iddq. اساساً از بخش ۲ تا ۵، این مقاله به چهار قسمت سازماندهی شده است که هر کدام یک زیرمجموعه از روش آزمون را توصیف می کند و نهایتاً بخش ۶ شامل نتیجه گیری ها می شود.

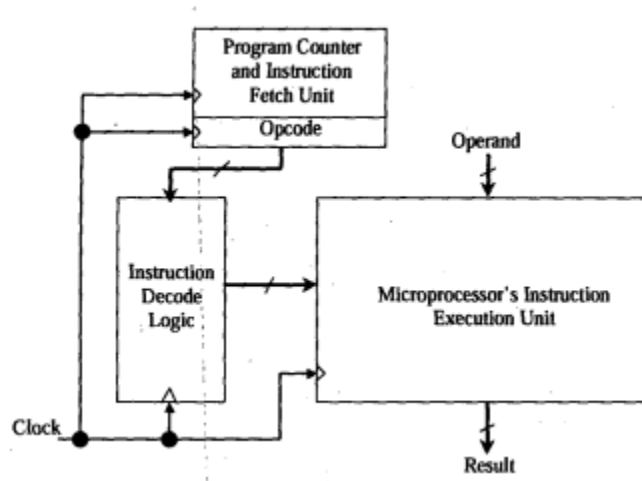
۲. آزمایش هسته میکروپروسور

تعداد زیادی از SOC شامل یک یا چند میکروپروسور/میکروکنترلر می شود. آزمایش میکروپروسورهای تعبیه شده نسبتاً پیچیده است. در کل، طراحی برای آزمون و طراحی های خودآزمون توکار مانند اسکن کامل، اسکن جزئی، بلوک های منطقی مختلف در یک میکروپروسور/میکروکنترلر. در این بخش، ما یک روش از نوع BIST را برای آزمایش هسته میکروپروسور استفاده شده در SOC توصیف می کنیم. این روش، صحت دستورالعمل ها را آزمایش می کند و بنابراین می تواند به عنوان یک روش آزمون وظیفه ای در نظر گرفته شود.

۲.۱ ساختار آزمون برای هسته میکروپروسور

ساختار گسترده یک میکروپروسور/میکروکنترلر در شکل 1a نشان داده شده است. همانطور که در این شکل نشان داده شده است، واحد دستورالعمل واکنشی شامل رمزالعمل دستورالعمل بعدی بر اساس آدرس در شمارنده برنامه می شود. این رمزالعمل توسط منطق کدگذاری دستورالعمل کدگذاری می شود که انتخاب وظیفه و سیگنال های کنترل

را برای واحد اجرا تولید می کند. بر اساس این سیگنال های کنترل، یکی از بلوک های منطق در واحد اجرا، وظیفه آن را محاسبه می کند. ابراند یا داده ها برای این محاسبه از حافظه سیستم به دست می آید.



شمارنده برنامه و واحد واکنشی دستورالعمل
 ابراند رمزالعمل
 واحد اجرای دستورالعمل میکروپروسسور منطق کدگشایی دستورالعمل
 نتیجه ساعت

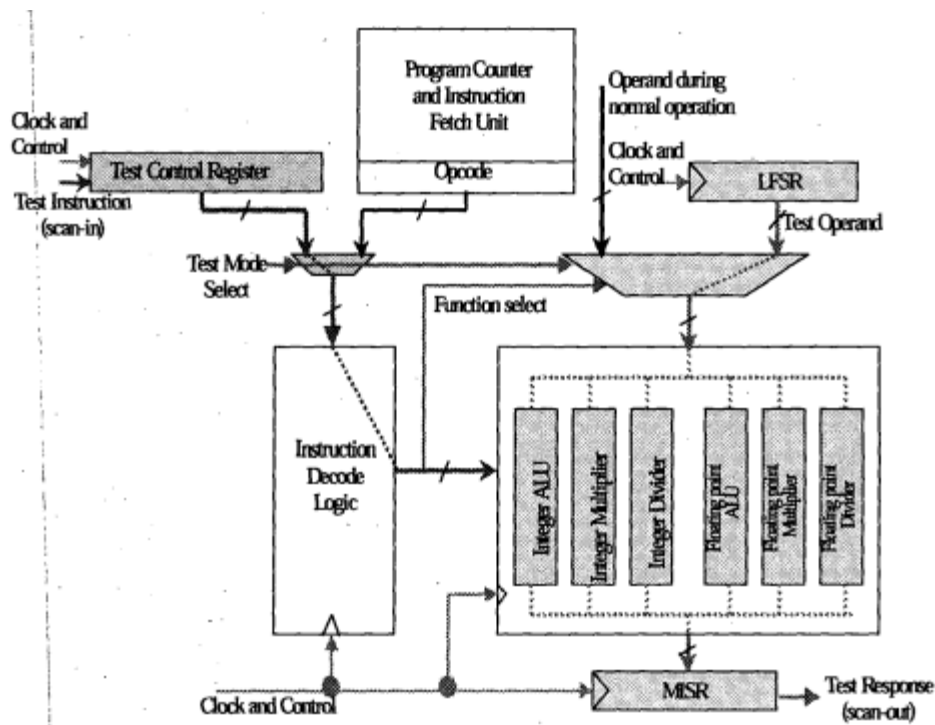
شکل 1(a): ساختار کلی یک میکروپروسسور

ساختار نشان داده شده در شکل 1a با اضافه نمودن سه ثبات به صورت نشان داده شده در شکل 1b اصلاح می شود. یک ثبات کنترل آزمون (TCR) برای فراهم نمودن رمزالعمل دستورالعمل های میکروپروسسور در مدت مد آزمون، یک ثبات شیفت فیدبک (LFSR) و دیگر ثبات شیفت فیدبک چند ورودی (MISR) برای تولید داده ها تصادفی و برای فشرده نمودن پاسخ آزمون به ترتیب. در مدت مد آزمون، داده ها از LFSR به عنوان ابراند برای دستورالعمل فراهم شده توسط TCR استفاده می شود. نتیجه محاسبه شده در MISR ذخیره می شود. باید توجه شود که به صورت نشان داده شده در شکل 1b، واحد اجرا شامل گروهی از بلوک های پیچیده اجرا کننده عملیات های منطقی و محاسباتی نقطه شناور و عدد صحیح می شود.

۲,۲ توالی آزمون برای هسته میکروپروسور

توالی آزمون به صورت زیر است:

۱. فعالسازی مد آزمون. در این مد، محتوای TCR به عنوان یک دستورالعمل استفاده می شود نه رمزعمل از واحد واکنشی دستورالعمل
۲. آغاز TCR، LFSR و MISR توسط سیگنال های کنترل آزمون.
۳. بارگذاری سریال TCR با رمزعمل برای یک دستورالعمل



اپراند در مدت عملیات عادی شمارنده برنامه و واحد واکنشی دستورالعمل ساعت و کنترل
LFSR ساعت و کنترل رمزعمل ثبات کنترل آزمون دستورالعمل آزمون (اسکن داخلی)
اپراند آزمون انتخاب وظیفه انتخاب مد آزمون
درایور نقطه شناور مالتی پلسر نقطه شناور ALU نقطه شناور تقسیم کننده نقطه شناور ضرب کننده عدد
صحیح ALU عدد صحیح منطق کدگشایی دستورالعمل
پاسخ آزمون (اسکن خارجی) MSRT ساعت و کنترل

شکل 1(b): پیاده سازی روش نوع BIST برای آزمون وظیفه ای هسته میکروپروسور

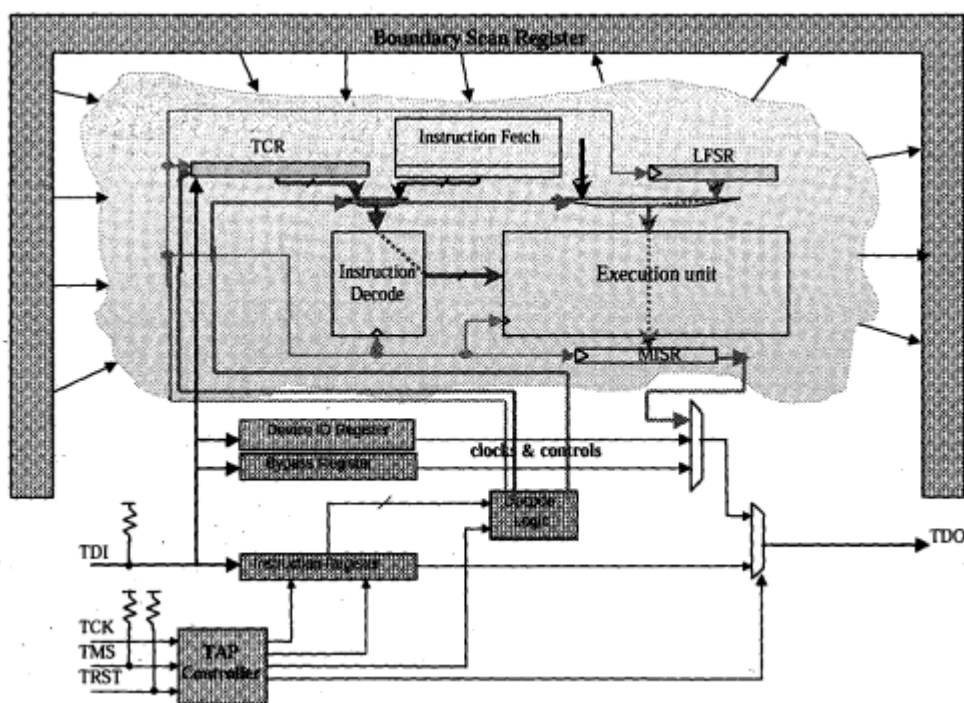
۴. LFSR و MISR ساعت برای یک تعداد چرخه. این مرحله به طور مکرر، یک دستورالعمل را در TCR با داده های LFSR اجرا می کند. به طور مثال، زمانی که ۱۰۰۰ چرخه استفاده می شوند، این دستورالعمل در TCR اجرا می شود (۱۰۰۰ بار - نهفتگی بلوک) با اپراندهای مختلف، داده های تصادفی فراهم شده توسط LFSR.

۵. اتخاذ سریالی محتوای MISR برای تعیین عبور/عدم موفقیت

۶. مقایسه محتوای MISR با یک امضای شبیه سازی از پیش محاسبه شده برای تعیین اینکه آیا یک خطا وجود دارد یا نه. تجهیزات خودکار آزمون (ATE) این مقایسه را انجام می دهد.

۷. مراحل #2 تا #6 را با دستورالعمل های مختلف تا زمانی که تمام دستورالعمل ها اجرا شوند تکرار نمایید.

کنترل این طرح، از طریق کنترلر TAP اسکن مرزی IEEE 1149.1 [6] انجام می شود. سیگنال های کنترل آزمون و پاسخ های آزمون از طریق آن عبور داده می شوند و توسط کنترلر TAP اسکن مرزی کنترل می شوند. شکل ۲، پیاده سازی کلی را نشان می دهد.



شکل ۲. پیاده سازی روش نوع BIST برای هسته میکروپروسسور

باید توجه نمود که در این طرح، محک آزمون شبیه سازی به عنوان داده های طلایی استفاده می شود که شامل تمام دستورالعمل ها با توالی LFSR و امضاهای MISR بعد از هر اجرا می شود. بنابراین، محتوای MISR عاری از خطا بعد از هر اجرای دستورالعمل برای مقایسه ساده در مدت آزمون تولید به عنوان استقرایی شناخته می شود. رویه بالا تعیین می کند که هر دستورالعمل به طور صحیح اجرا می شود و از اینرو پوشش وظیفه ای را فراهم می کند. همچنین برای تخمین پوشش خطا در یک بلوک فردی برای تقریباً ۱۰۰۰ اپراند تصادفی، شبیه سازی خطا در هر اجرای ثبت شده انجام می شود. برای شبیه سازی خطا، یک ابزار EDA تجاری استفاده می شود.

۳. آزمایش حافظه های تعبیه شده

زمانی که آزمون هسته میکروپروسور تکمیل می شود، این هسته برای آزمایش حافظه تعبیه شده و دیگر هسته های روی تراشه استفاده می شود. در این بخش، ما روش خود را برای آزمایش حافظه تعبیه شده با استفاده از قدرت محاسبه هسته میکروپروسور توصیف می کنیم. در این روش، یک برنامه زبان اسمبلی روی هسته میکروپروسور اجرا می شود که الگوهای آزمایش حافظه را تولید می کند. هرچند، ما از الگوریتم March استفاده می کنیم، هر الگوریتم آزمایش حافظه می تواند برای این مقصد استفاده شود. پاسخ آزمون نیز توسط هسته میکروپروسور ارزیابی می شود.

۳.۱ رویه آزمون

رویه زبان اسمبلی نمونه با استفاده از الگوریتم شکل ۳ نشان داده شده است. این مثال از عملیات های خواندن/نوشتن با داده های ۰۱۰۱۰۱۰۰۰۰۰۰۱ در مرتبه افزایشی و ۱۰۱۰۱۰۰۰۰۰۱۰ در مرتبه کاهشی استفاده می کند. این آزمون برای یک RAM ۲ مگابیتی سازماندهی شده به صورت 16K*16 است.

/* رویه ای برای آزمایش RAM تعبیه شده */

/* A0 شماره آدرس، D0 شامل داده های آزمون (نوشتن)، D1 برای خواندن داده ها (پاسخ)؛ A0, D0-D2 ثبات

های مقصد عمومی است */

/* آغاز */

/**آغاز شمارنده آدرس**/
/**این داده های آغاز است**/
/**این برای پاکسازی کلمه حافظه بعد از خواندن استفاده می شود**/

/**رویه آزمون**/
آغاز

/**این مقدار در D0 به موقعیت آدرس داده شده توسط A0 نوشته می شود**/

/**FFFFH آخرین آدرس است**/

/**حافظه آغاز می شود، آزمون راه اندازی**/

/**خواندن/نوشتن در مرتبه افزایشی**/

/**این داده آزمون (۱۰۱...۰) است**/

/**این عملیات نوشتن است**/

/**این عملیات خواندن است**/

/**این کار کلمه حافظه را پاک می کند**/

/**داده های خواندن 5555h نیست**/

/**آخرین آدرس**/

/**خواندن/نوشتن در مرتبه کاهشی**/

/**این داده آزمون (۱۰۱...۰) است**/

/**این عملیات نوشتن است**/

/*این عملیات خواندن است*/

/*این کار کلمه حافظه را پاک می کند*/

/*داده های خواندن AAAAH نیست*/

/*0000H آخرین آدرس است*/

شکل ۳: یک برنامه زبان اسمبلی برای الگوی March گسترده اجراشونده روی هسته میکروپروسور.

برنامه زبان نشان داده شده در شکل ۳، توسط اسمبلر هسته میکروپروسور تبدیل به دودویی می شود. کد دودویی اسمبلر به روش مشابه با بردارهای آزمون از ATE برای هسته میکروپروسور استفاده می شود. داده های تامین شده برای هسته میکروپروسور به طور مستقیم از آزمونگر می آید. در برخی موارد، در مدت این آزمون، آزمونگر تبدیل به حافظه سیستم برای هسته میکروپروسور می شود. زمانی که داده ها از آزمونگر، دستورالعمل هایی میکروپروسور به شکل دودویی هستند، هسته میکروپروسور عملیات ها را به صورت هدفمند توسط دستورالعمل برنامه اجرا می کند. خروجی نهایی از میکروپروسور به آزمونگر، یک وضعیت عبور/رد است. باید توجه نمود که برنامه آزمون نشان داده شده در شکل ۳، به محض اینکه یک خطا رخ دهد متوقف می شود (نوشتن دچار خرابی می شود). آزمونگر این وقوع خرابی را ذخیره می کند و از اینرو موقعیت بیت-خراب بدون هر تلاش اضافی مشخص می شود.

این روش سراسر است و عمدتاً الزامات ATE را ساده می کند. برای آزمون سرعت، هیچ جریمه ای را برای عملکرد ایجاد نمی کند و هر الگوریتم آزمون حافظه می تواند استفاده شود. تنها اشکال برای این روش اینست که نیاز به API ویژه دارد (واسطه برنامه کاربردی) تا اطلاعات تولید شده دودویی را توسط اسمبلر هدایت کند. این API برای آزمونگر، درک کد دودویی اسمبلر را در مدت توسعه برنامه آزمون و در مدت آزمون میسر می کند.

جالب توجه است که روش های مشابه در گذشته پیشنهاد شده اند که یک برنامه آزمون دقیق می تواند برخی از آرایه های حافظه را روی تراشه آزمایش نماید اگر کش دستورالعمل کاملاً در ابتدا آزمایش شود [7-9]. این روش توصیف شده در بالا اساساً از [7-9] با توجه به واسطه آزمونگر متفاوت است و نیاز به کش ۱ به خوبی شناخته دارد.

۴. آزمایش هسته های خاص تابع

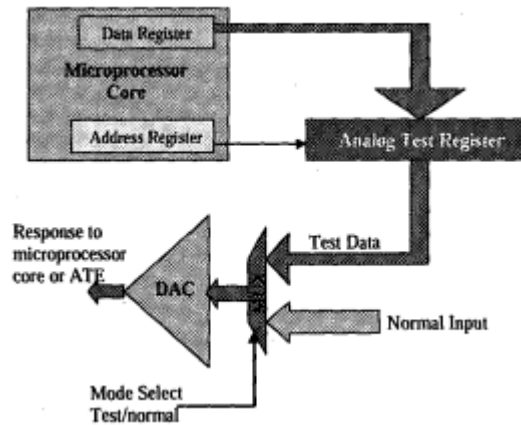
زمانی که هسته میکروپروسسور و حافظه های روی تراشه آزمایش می شوند، هسته میکروپروسسور برای آزمایش هسته های دیگر استفاده می شود. در این بخش، ما آزمایش مبدل D/A تعبیه شده را توصیف می کنیم (DAC).

۴.۱ رویه آزمون

مفهوم اصلی مشابه با بخش ۳ است. یک برنامه زبان اسمبلی توسعه یافته، توسط اسمبلر به شکل دودویی تبدیل می شود و روی هسته میکروپروسسور از طریق API آزمونگر اجرا می شود.

هرچند، برای حل مشکل کاربرد داده های آزمون از هسته میکروپروسسور به DAC، ما از یک ثبات اضافی (ثبات آزمون آنالوگ، ATR) اضافه می کنیم. در مد آزمون، محتوای ATR می تواند توسط آدرس دهی شاخص، مانند توسط آدرس دهی از طریق هر یک از ثبات های آدرس میکروپروسسور عوض شود. در مد آزمون، ATR، ورودی ها برای DAC توسط یک مالتی پلکسر فراهم می کند. در مدت مد نرمال، ATR توسط همان مالتی پلکسر قطع می شود. این مفهوم در شکل ۴ نشان داده شده است.

با کمک ATR، محرک آزمون تولید شده توسط هسته میکروپروسسور می تواند برای DAC اعمال شود. رویه ها برای تولید محرک آزمون در ولتاژ آفست (VOS)، گستره مقیاس کامل (FSR)، از دست دادن کدها و گذرهای عمده، غیرخطی دیفرانسیل (DNL) و غیرخطی انتگرال (INL) در زیر توصیف شده است:



ثبات داده ها

هسته میکروپروسسور

ثبات آدرس

ثبات آزمون آنالوگ

داده های آزمون DAC پاسخ به هسته میکروپروسسور یا ATE

ورودی نرمال آزمون انتخاب مد/نرمال

شکل ۴: مکانیزمی برای کاربرد داده های آزمون از هسته میکروپروسسور تا DAC

ولتاژ آفست (VOS): برای DAC، ولتاژ آفست، ولتاژ خروجی آنالوگ است زمانی که یک کد تمام صفر یا خالی در ورودی ها اعمال می شود. پهنای بردار آزمون همان پهنای بی تفکیک DAC است در حالیکه که طول 2^N مانند 16,32,64 و غیره است. علاوه بر اجتناب از اعوجاج ناشی از نویز، ما همان بردار را چندین بار اعمال می کنیم و متوسط خروجی برای محاسبه VOS اتخاذ می شود. بردار آزمون به سادگی با بارگذاری تمام صفرها به یکی از ثبات های داده میکروپروسسور به دست می آید. یک دستورالعمل میکروپروسسور برای این هدف استفاده می شود، MVI (Ai) 0000H که در آن (Ai) ثبات آدرس نام است.

گستره مقیاس کامل (FSR): گستره مقیاس کامل، تفاوت بین ولتاژ خروجی آنالوگ است زمانی که

for embedded DRAM", IEEE J. Solid State Circuits, vol. 33(11), pp. 1731-1740, Nov. 1998.

9. J. Saxena, P. Ploicke, K. Cyr, A. Benavides and H. Malpass, "Test strategy for TI's TMS320AV7100 device", IEEE Int. Workshop on Testing Embedded Cores based Systems, pp. 3.2.1-3.2.6, 1998.
10. R. Rajsuman, "A new test method for testing embedded memories in core based system-on-a-chip ICs", IEEE Int. Workshop on Testing Embedded Cores based Systems, pp. 3.4.1-3.4.6, 1998.
11. F. Zarrinfar and R. Rajsuman, "Automated Iddq testing from CAD to manufacturing", IEEE Int. Workshop on Iddq Testing, pp. 50-55, 1995.
12. M. Colwell, R. Rajsuman, Z. Sarkari and R. Abrishami, US patent No. 5,644,251, July 1, 1997; and US patent No. 5,670,890, Sep. 23, 1997.
13. R. Rajsuman, "Design-for-Iddq-Testing for embedded cores based system-on-a-chip", IEEE Int. Workshop on Iddq Testing, 1998.

References

1. D&T Roundtable, "Testing embedded cores", IEEE Design and Test of Computers, pp. 81-89, April-June 1997.
2. J. Hutcheson, "Executive advisory: the market for systems-on-a-chip", and "The market for systems-on-a-chip testing", VLSI Research Inc.
3. IEEE P1500 CTAG web page.
4. Y. Zorian, "Test requirements for embedded core based systems and IEEE P1500", IEEE Int. Test Conf., pp. 191-199, 1997.
5. VSI Alliance, Manufacturing related test development WG specifications, 1998.
6. IEEE Standard 1149.1, "IEEE standard test access port and boundary scan architecture", IEEE Press, 1990.
7. D. K. Bhavsar and J. H. Edmondson, "Testability strategy of the alpha AXP 21164 microprocessor", IEEE Int. Test Conf., pp. 50-59, 1994.
8. J. Dreibelbis, J. Barth, H. Kalter and R. Kho, "Processor based built in self test

این مقاله، از سری مقالات ترجمه شده رایگان سایت ترجمه فا میباشد که با فرمت PDF در اختیار شما عزیزان قرار گرفته است. در صورت تمایل میتوانید با کلیک بر روی دکمه های زیر از سایر مقالات نیز استفاده نمایید:

لیست مقالات ترجمه شده ✓

لیست مقالات ترجمه شده رایگان ✓

لیست جدیدترین مقالات انگلیسی ISI ✓

سایت ترجمه فا ؛ مرجع جدیدترین مقالات ترجمه شده از نشریات معتبر خارجی