

デジタル・パターン・ジェネレータ — デジタル・システム開発のための必須の機器

組み込みシステムのテスト及びデバッグのために使用される信号ジェネレータの世界で、生成された信号の特徴と彼らを生成する目的に従って、我々は、機器の3つのカテゴリーを見つけることができます。‘組み込みシステム’は、その広義で理解されるべきです、それは、アナログ、デジタルまたはアナログ/デジタル混合のシステムを含むマイクロプロセッサあるいはマイクロコントローラ、ASIC、FPGA、SoC(システム・オン・チップ)、DSP、といった‘PCからの’コンピュータ・リソースを持つ電子システムです。概要に関して以下のテーブルを参照してください。

機器タイプ	信号特性	使用法
デジタル・パターン・ジェネレータ	デジタル。 LVCMOS、LVTTTL、LVDS...といったデジタルの標準電圧に従った‘1’と‘0’の連続。	主として機能的検証。 生成された信号は、デジタル・システムへの入力としてデジタル情報を伝達するために生成されます。大概、最低16チャンネル以上の機能。
パルス/パターン・ジェネレータ	デジタル、一連の矩形パルス。 立上り/立下り時間、信号ドライブ力、ジッター...といった、信号の電気特性をチューニングする能力。	電気的な入力バッファのテストと認証。 デジタル・パターン・ジェネレータのように機能的検証のために時々使われま す。大概、1あるいは2チャンネルのみの機能。
ウェーブフォーム・ジェネレータ	アナログ。 出力の振幅を定義する能力で、正弦波、のこぎり波、矩形波あるいは、任意の複 合信号のような標準波の生成。	システムのアナログ入力をステイミュ レートする。 この様な入力の電気検証とアナログ入 力を主とした複合システムの機能的検 証のために使用されます。

Table 1: Signal generators typology

この文書で、組み込みシステム・デザインにかかわっているすべてのエンジニアのために、我々は、必須の一個の機器として‘デジタル・パターン・ジェネレータ’を明確に喚起するでしょう。

デジタル・パターン・ジェネレータは、デザイン中に半導体やデジタル・システムを改善します。

デザイン中に、デジタル・パターン・ジェネレータは、ほとんどすべてのタイプのデジタル・デバイスのために必須のステイミュラス・ソースです：デジタルや混合信号のASIC、FPGA、マイクロプロセッサとマイクロコントローラ。デジタル・パターン・ジェネレータは、機能的なテスト、新しいデザインや既存のデザインの障害分析のデバッグに役立ちます。

デジタル・パターン・ジェネレータは、まだ利用できないシステム・コンポーネントの代わりになるために、デザイン・サイクルの早期で使うことができます。例えば、普通に信号を提供するだろうプロセッサがまだ存在しないときに、デジタル・パターン・ジェネレータは、新たに開発されたバス回路へ割込みやデータを送るために、プログラミングされるでしょう。

それは、新しいテストケースを探索することを可能にします—そして、コードが動作し、そして十分に堅牢なことを確かめるのを助けるために稀に遭遇したテスト条件を作り出します—そして、これは完全なハードウェアが、利用できる以前でさえ。

デジタル・パターン・ジェネレータは、望む状態へ回路を置き、それからフル・スピードでそれを操作させるか、あるいは一連の状態を通して回路をステップするために使用することもできます。正しい特定の初期化シーケンスを生成し、そして、回路のオペレーションの多くのモードの探査で、コントロール・レジスタを構成することが役立つでしょう。今日では、回路は、数十メガヘルツ以上で走るかもしれないデジタル・インターフェースを使用するので、GPIOで連結されたマイクロコントローラーでそれをやる古い方法は、これ以上十分ではありません。

デジタル・ロジック信号を生成するときに失った部品や最大限の柔軟性を提供することの代わりになることによって、デジタル・パターン・ジェネレータは、新製品のタイム・トゥ・マーケットを加速するでしょう。デジタル・パターン・ジェネレータは、まさにdevice under test(DUT)をスティミュレートするために複雑なデジタル・ビット・ストリームが、必要とされるどこでも効果的な解決策です

組込みシステムは、ワークステーションでデザインされます。

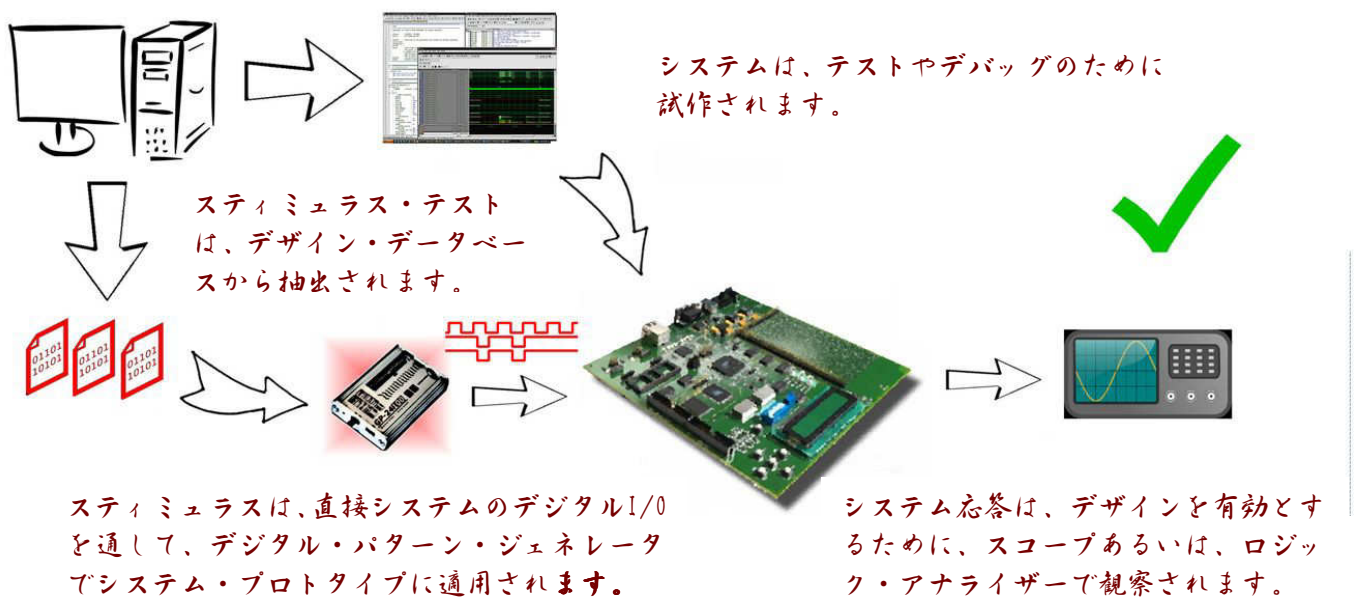


Figure 1: Principle of stimulus-and-response setup for a prototyped digital system

‘デバッグ&検証’は、‘測定のみ’を意味しません。

Byte Paradigmは、組込みハードウェアとソフトウェア・デザインにかかわっている300以上の回答者へ2009年に調査を行いました。この調査は、以下を示します：

-エンジニア(61%)の大多数が、‘ハードウェア・プロトタイプを使用することが、組込みシステム・デバッグをスピードアップする’という意見に同意します。全体の回答者(合計335)のほんの10%は、この意見にいくぶん異なるか、あるいは、異なります(table 2参照)。

-総回答者の83%以上は、スティミュラスの生成が少なくとも‘システム応答観察に必要なこと’を言明します。それがIP、FPGA、ASIC、SoCまたはフル・組込みシステムテストに関わるかどうかにかかわらず、図は、道理にかなって同じです(平均の見解を提供するfigure 2参照)。

	同意	やや同意	どちらかと言えば反対	反対	分らない	回答数
ハードウェア・プロトタイプを使用することが、組込みシステム・デバッグをスピードアップする	61.2% (205)	29.3% (98)	2.7% (9)	1.8% (6)	5.1% (17)	335

Table 2: 2009 survey results (excerpt 1)

彼らがデザインしているものをテストし、デバッグするために、開発エンジニアの大半は、‘実際のハードウェア’プロトタイプに行き着きます。基本的に、‘プロトタイプでテストすること’は、いつも‘スティミュレートして観察する’に還元されます。

興味深いことにこの調査は、エンジニアが大概オシロスコープ(回答者の69.5%)、ロジック・アナライザ(回答者の57.1%)、そして、JTAGプローブ(回答者の59.3%)を十分に装備していることも示します。

一方で、彼らがテスト及びデバッグ作業を行うために、回答者の20.6%しか、ラボでデジタル・パターン・ジェネレータを見つけません。この数字は、ウェーブフォーム・ジェネレータ・サイン・ウェーブといったアナログ信号生成についていえば32.8%に上昇します。この状況について尋ねられたときに、暫定結果は、エンジニアがより市販の標準を使うことを熱望するだろうことを示します。将来のプロジェクトに対するスティミュラス生成のために利用できる信号ソース。振り返ってみると、もし機能的なテスト及び彼らの最新のデジタル・システムのためにデジタル・パターン・ジェネレータを使うなら、彼らは貴重なデザイン時間の節約になっただろうことにも同意します。

エンジニアが系統的にこのようなツールとところが、彼らはいつもPCとオシロスコープを受入れます。装備していない理由は、あまり明らかではありません。この質問は、Byte Paradigmによって調査されています。

すべてのエンジニアは、デジタル・パターン・ジェネレータを有するべきです。

Byte Paradigmにおいて、我々は、デジタル・パターン・ジェネレータが組込みシステムのテスト、デバッグ及び検証の多くの挑戦に打ち勝つのを助けると思っています。

我々は、システム・デバッグをスピードアップするデザイン・サイクルの初期のプロトタイプをするエンジニアの意見を共有し、そして、その結果として、全般的な製品デザイン・サイクルを短くするのを助けます。プロトタイプを作られた制度を試すことは、2つのタイプの職務を基本的に必要とします：1) システムへの入力スティミュラスを生み出すこと、そして、2) システムの応答を分析すること。

‘スティミュラスと一応答’の挑戦を解決するために、もし彼が正しく装備すれば、エンジニアは、貴重なデザイン時間を節約するでしょう。スコープ、ロジック・アナライザ及びたぶんより専門化したアナライザが‘分析ジョブ’をするのを助けるだろうといういかなる疑いもありません。彼らは、デジタル・システムの重要な友です：デジタル・パターン・ジェネレータ。

When testing an embedded system, what is the most challenging?

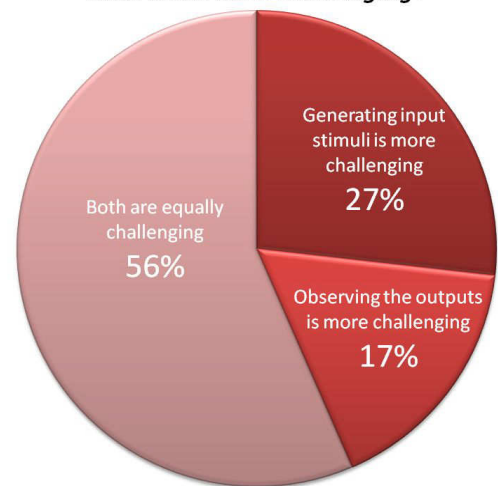


Figure 2: 2009 survey results (excerpt 2)

About the author

Frederic Leens is Sales and Marketing Manager at Byte Paradigm.

He can be reached at: frederic.leens@byteparadigm.com