

Niosの活用法

浅井 剛, 飛永 徹, 中根隆康

ここでは、ソフト・マクロのCPUコアを使ったFPGAの設計法と、その上で動作するソフトウェアの開発法について解説します。FPGA設計におけるソフト・マクロのCPUの活用は、いまや珍しいことではなくなりました。ソフト・マクロのCPUには、ディスコン(製造中止)の心配がありません。また、中・大規模FPGAであればユーザ論理の“すき間”に収まるので、部品点数の削減につながります。

5-1 ソフト・マクロのCPUをFPGAに組み込もう！

ソフト・マクロCPUとそのハードウェア開発環境 浅井 剛

本稿では、米国Altera社が提供している32ビットRISC (reduced instruction set computer) プロセッサ・コア「Nios」の概要を解説します。Niosは、2000年6月に発表されました。本稿執筆時(2003年12月)における最新版はバージョン3.1です。

のCPUコアです。RISCアーキテクチャをとり、32ビットと16ビットの2種類があります。周辺機能のカスタマイズも可能です。

図1にNiosを構成する機能モジュールを示します。NiosにはCPUのみならず、メモリや周辺回路まで含まれていません。これらは「Avalon」という独自のオンチップ・バスで接続されます。ユーザ論理は、バス・ブリッジを介してAvalonバスに接続します。機能的には、半導体メーカーが出荷しているマイコンと変わりません。

1. Niosのアーキテクチャ

NiosはAltera社のFPGA向けに特化したソフト・マクロ

●CPUコアの特徴

CPUコアは、キャッシュ・メモリやJTAGデバッガをサポートしています。これらは、バージョン3.0からサポートされた機能です。

表1に示すように、CPUコアには32ビット版と16ビット版の2種類があります。一般に、処理性能を求める場合は

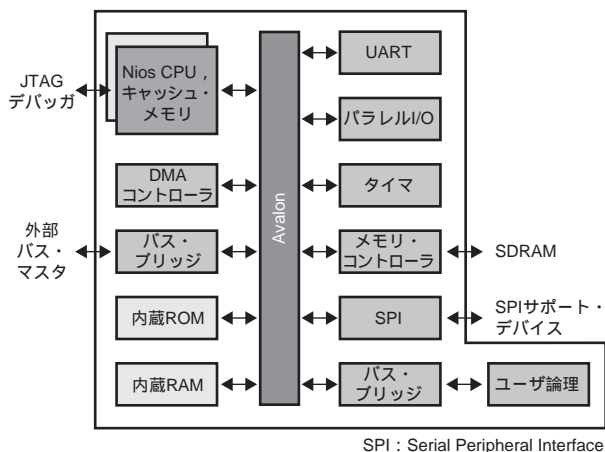


図1 Niosを構成する機能モジュール

CPU、メモリ、周辺回路が含まれている。これらの機能モジュールはAvalonバスで接続される。

表1 Nios プロセッサ・コアの概要

機能	32ビット版	16ビット版
データ・バスの幅(ビット)	32	16
論理演算ユニット(ALU)の幅(ビット)	32	16
内部レジスタの幅(ビット)	32	16
アドレス・バス・サイズ(ビット)	32	16
命令コード長(ビット)	16	16
ロジック・エレメント(LE)数(標準)	1,500未満	1,100未満
最大動作周波数(f_{MAX})	125MHz以上	125MHz以上

32ビット、回路規模を抑えたい場合は16ビットと使い分けられているようです。ただし、回路規模の差は約400LE(Logic Element)しかありません。このため、32ビット版を採用するケースが多いようです。

Niosのアーキテクチャの特徴は、以下の3点であると筆者は考えています。

- ◆ カスタム演算命令追加機能
- ◆ レジスタ・ウィンドウ
- ◆ Avalonバス

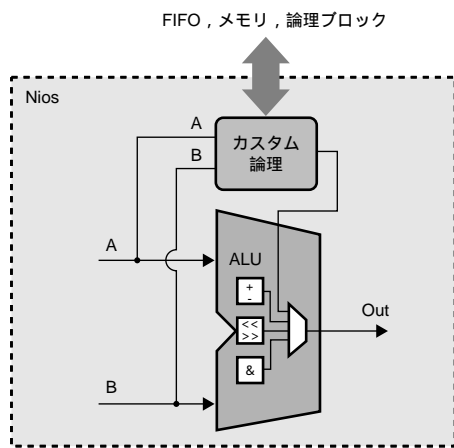


図2 カスタム演算命令の追加

独自に設計した演算モジュールをCPUコアに組み込むことができる。命令セットを拡張し、専用の命令で呼び出す。

●カスタム演算命令追加機能

最大で5個まで、カスタム命令を追加することができます。独自に設計した演算モジュールをCPUコアに組み込み、命令セットを拡張します(図2)。

こうすることで、ソフトウェアでは時間がかかるひとまじりの演算が1命令で実行可能となります。繰り返し実行される演算に適用すれば、かなりの性能向上を図れるでしょう。ハードウェアで設計した機能ブロックを、ソフトウェアから専用命令で呼び出すことができれば、所望のシステムをシンプルに実現できるかもしれません。

●レジスタ・ウィンドウ

Niosでは、128個、256個、512個の3種類の総レジスタ数を選択できます。CPUのレジスタは、FPGAが内蔵するメモリ・ブロックを使用します(LEで実現することも可能)。このため、多くのレジスタを用意することができるでしょう。

128レジスタ時のCWP(Current Window Pointer)値とレジスタ・ウィンドウの関係を図3に示します。すべてのレジスタは、CWPというポインタ・レジスタで管理されます。ソフトウェアからは、32個のレジスタが1ウィンドウとして見えます。ウィンドウは16レジスタ単位に移動します。前のウィンドウと重複する16個のレジスタは、共有エリアになります。例えば、サブルーチン・コールにおいて引き数や戻り値を格納するために使用します。多くのマ

内部レジスタ・ファイル	CWP=6 (上限)						
Reg[120..127]	%i0..%i7						
Reg[112..119]	%L0..%L7	CWP=5					
Reg[104..111]	%o0..%o7	%i0..%i7					
Reg[96..103]		%L0..%L7	CWP=4				
Reg[88..95]		%o0..%o7	%i0..%i7				
Reg[80..87]			%L0..%L7	CWP=3			
Reg[72..79]			%o0..%o7	%i0..%i7			
Reg[64..71]				%L0..%L7	CWP=2		
Reg[56..63]			%o0..%o7	%i0..%i7			
Reg[48..55]				%L0..%L7	CWP=1		
Reg[40..47]			%o0..%o7	%i0..%i7	CWP=0 (下限)		
Reg[32..39]				%L0..%L7	(アンダフローまたはトラップ)		
Reg[24..31]			%o0..%o7	%i0..%i7			
Reg[16..23]				%L0..%L7			
Reg[8..15]			%o0..%o7				
Reg[0..7]	%g0..%g7	%g0..%g7	%g0..%g7	%g0..%g7	%g0..%g7	%g0..%g7	%g0..%g7

図3 128レジスタ構成時のレジスタ・ウィンドウ

32個のレジスタが1ウィンドウになる。ウィンドウは16レジスタ単位で移動する。前のウィンドウと重複する16個のレジスタは、共有エリアとして引き数や戻り値を格納するために使用する。

CWP=0のレジスタ・グループ	
%r24..%r31	aka %i0..%i7
%r16..%r23	aka %L0..%L7
%r8..%r15	aka %o0..%o7
%r0..%r7	aka %g0..%g7