

組込みシステム向けにSDRAMを効率的に活用するベクトルメモリアクセス機構の設計

岸本進太郎[†] 大津 金光[†] 横田 隆史[†] 小島 駿[†]
[†] 宇都宮大学工学部情報工学科

1 はじめに

近年、組込みシステムには FPGA(Field Programmable Gate Array) の利用が増加している。FPGA とはユーザーによって自由に回路を構成することのできる LSI であり、ユーザーが求める処理をハードウェアロジックとして実装することで用途に合わせ最適化した専用回路の開発が可能である。最近では、FPGA の高性能化と大容量化に伴い、大規模回路を構成することが可能となってきた。これらの背景により、組込みシステムでの FPGA の利用増加が期待されている。[1] 専用回路は高性能だが開発コストが大きく、開発コスト削減のため専用回路の一部の処理を担当させるソフトコアプロセッサの実装が考えられる。ソフトコアプロセッサとは FPGA 上に構成するプロセッサのことである。低コストのソフトコアプロセッサを高性能化し、性能を専用回路に近づけることで全体としての開発コスト削減につながる。そこで、我々はスケーラブルなベクトル拡張を行ったソフトコアプロセッサを開発している [1]。一般的に、組込みシステムではコストが優先されるためメモリシステムの転送性能が高くないことが想定され、そのような状況で効率的なベクトル処理を実現するためには、SDRAM の転送性能を十二分に活用することが必須である。図 1 に本稿で検討するメモリアクセス機構の構成図を示す。プロセッサと SDRAM のアクセスにはメモリコントローラを仲介し、アクセスの制御を行う。また、プロセッサと SDRAM の動作周波数が異なるためメモリコントローラ内部に非同期 FIFO を用いる。

2 マルチバンクアクセスによる効率的なデータ転送

メモリアクセス先となる SDRAM は行・列による 2 次元配列構造のバンクを複数備える構造をとる。連続でメモリアクセスを行う場合のアクセスパターンは次の 3 パターンが考えられる。[2]

1.” 同一バンクの異なる行への連続アクセス”

このアクセス方法では、次のアクセスが可能となるのは前のアクセスしたデータがデータバス上に読み出された後になるためデータバスの利用率が低い。

2.” 同一バンクの同一行への連続アクセス”

このアクセス方法では、読み出し対象となる行が行バッファに一定期間残るため行アドレスの指定を省略

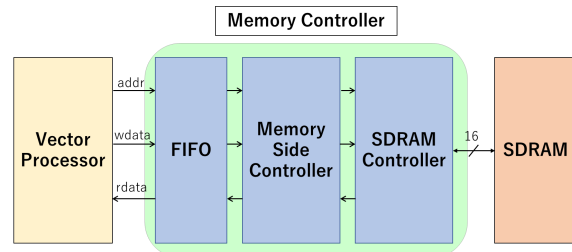


図 1: システム構成

し、読み出すデータのバス上での出現間隔を最短 0 にすることが可能になる。しかし、連続アクセスが可能な回数は行サイズに依存するためデータバス利用率の向上には繋がりにくい。

3.” 異なるバンクへの連続アクセス”

SDRAM の備える複数バンクはそれぞれ独立してメモリアクセスを処理可能な性質により、異なるバンクへの連続したメモリアクセスは直前のアクセスの終了を待たずに、別のバンクに対するアクセスを開始される。先ず、同一バンクに対する連続アクセスでは最初のアクセス要求からそのデータがデータバス上に出現するまでの待機時間 T_A が経過したのちに次のアクセスを開始され、データバス上にデータが出現する間隔も T_A 以上となる。一方で、異なるバンクへの連続アクセスでは最初の待機時間 T_A の間に次のアクセスを開始できる。それにより、データバス上に要求データが出現する間隔は最初のアクセス開始時から次のアクセス開始時までの間隔となり、これは T_A より小さく、データバスの利用率が高くなる。本稿ではこの 3 番目のメモリアクセス方法をマルチバンクアクセスと呼称する。

図 2 にマルチバンクアクセス時の信号入力タイミングの様子を示す。SDRAM のアドレス指定はバンクアドレスと各バンクの行アドレスの指定、バンクアドレスとバンクの列アドレスの指定の 2 ステップにより行われる。addr は行または列のアドレスの信号、bank はバンクアドレスの信号、ras は 0 の時に addr を行アドレスとして示す信号、cas は 0 の時に addr を列アドレスとして示す信号であり、data は書き込みまたは読み出しデータを示す信号である。図 2 のように、マルチバンクアクセスでは data 信号に bank1data が出現するよりも前に bank2 にアクセスするためのアドレス指定が先行する。

本研究では、ベクトルロード・ストアのためのメモリアクセスを 3 番目のアクセスパターンを利用し、FPGA

Design of vector memory access mechanism to efficiently utilize SDRAM for embedded systems.

[†]Shintaro Kishimoto, Kanemitsu Otsu, Takashi Yokota and Shun Kojima

Department of Information Science, Faculty of Engineering, Utsunomiya University (†)

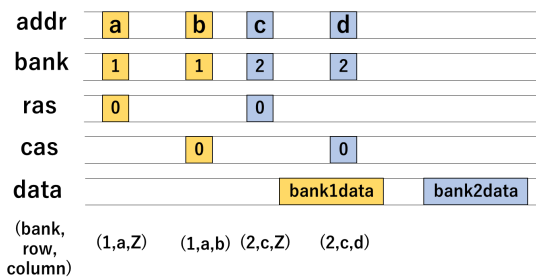


図 2: マルチバンクアクセスタイミングチャート

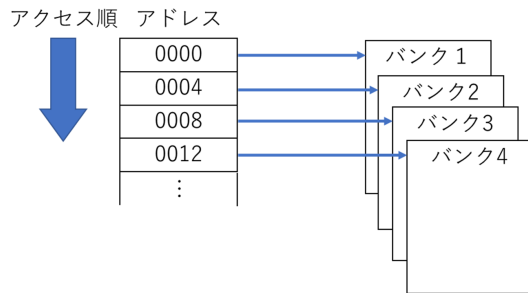


図 3: 連続したアドレスとバンクの関係例

と SDRAM 間のデータ転送を効率化する。ベクトル命令によって連続して発行されるメモリアクセスは局所性の原理により、連続したメモリ領域への読み書きを順に行う場合が多い。この特徴を利用し、図 3 に示すように連続したアドレスが異なるバンクになるようなアドレスの割り当てを行う。

3 SDRAM とのアドレス対応

マルチバンクアクセスによるメモリアクセスの効率化を図るにあたり、SDRAM へ送信するアドレスの制御を図 1 の SDRAM コントローラで行う。SDRAM 側で処理するデータのアドレスとプロセッサから送るデータのアドレスの対応関係を図 4 に示す。プロセッサ側では 1 アドレスが 8bit データを指し、SDRAM 側では 16bit が対応することを考え、プロセッサから送るアドレスの下位 1bit は使用しない。実装を検討している SDRAM の容量は 512MB であり、アドレスの指定に 29bit が利用され、そのうち行アドレス 15bit、列アドレス 10bit、バンクアドレス 3bit を割り当てる。

[27:13]bit の 15bit を行アドレス、[9:3][2:0]bit の連結を列アドレスの指定に使用する。バンクアドレスは [5:3]bit に割り当てる。下位 3bit ではなく [5:3]bit に割り当てることにより、マルチバンクアクセスが可能となる。それは SDRAM の機能としてバーストアクセスというものがあり、1 度のメモリアクセス要求で 16bit 単位の 8 回連続アクセス、すなわち 128bit 連続領域へのメモリアクセスが行われるためである。128bit 毎に読み出し、書き込みが行われるため、次に必要なアク

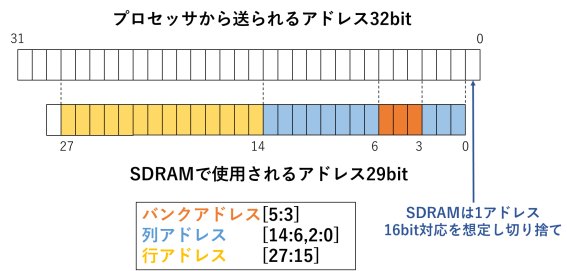


図 4: アドレス対応図

セス先のアドレスは直前のアドレス+8 (=128 ÷ 16) であり、これは図 3 で示したバンクアドレスのビットが変化することを意味する。アクセス先のバンクが異なるため次のアクセスではマルチバンクアクセスが利用可能となる。

本研究で対象とするソフトコアプロセッサでサポートするベクトルメモリアクセス命令はベースアドレスとなるスカラレジスタの値にオフセットを加えることで次のアクセス先のメモリアドレスが算出され、それをもとに順番にメモリアクセス命令を発行する。アクセス単位は 8bit, 16bit, 32bit, 64bit のいずれかであり、本稿ではまず連続したメモリ上に配置されるデータへのベクトルロード命令の実装から着手する。

本稿で提案するベクトルロード命令に合わせたマルチバンクアクセスの活用法は、 $128 \div (\text{アクセス単位}) = X$ として、最初のアドレス+X を次のアクセス先とする。これにより、各アクセス単位での同一バンクに対する連続アクセスを避け、SDRAM のバースト長に合わせたマルチバンクアクセスとなる仕様を現在検討中である。

4 おわりに

本稿では、連続なベクトルメモリアクセス実行時のボトルネックとなりがちなデータバス幅の有効活用方法を、SDRAM のメモリアクセスの特性を利用したアドレスの対応関係として検討した。

謝辞

本研究は、一部 JSPS 科研費 20K11726 の援助による。

参考文献

[1] Yoshiki Kimura, Kanemitsu Ootsu, Tatsuya Tuchiya, Takashi Yokota: “Development of RISC-V Based Soft-core Processor with Scalable Vector Extension for Embedded System”, Proceedings of the The 8th International Virtual Conference on Applied Computing & Information Technology June 2021 Pages 13-18

[2] Tomoya Kikuchi, Yoshiki Kimura, Kanemitsu Ootsu, Takashi Yokota: “Development of Soft-Core Processor with Efficient Array Data Transfer Mechanism”, Eighth International Symposium on Computing and Networking Workshops 2020