

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】<http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。



設計マニュアル（暫定）

CB-9 ファミリ VX/VM タイプ

0.35 μm CMOS セルベース IC (CBIC)

NB85E, NB85ET 編

資料番号 A14335JJ3V1DM00 (第3版)

発行年月 January 2002 NS CP(N)

© NEC Corporation 1999

[メモ]

目次要約

第 1 章	概 説	... 16
第 2 章	クロック制御回路の接続	... 19
第 3 章	VFB/VDB への ROM/RAM の接続	... 23
第 4 章	VSB への接続	... 40
第 5 章	メモリ・コントローラ (MEMC) の接続	... 56
第 6 章	NPB への接続	... 62
第 7 章	キャッシュの接続	... 69
第 8 章	インサーキット・エミュレータ (IE) との接続	... 76
第 9 章	注意事項	... 82
第 10 章	テスト回路設計	... 91
第 11 章	トータル・チップ・シミュレーション	... 106
第 12 章	ROM コードの作成	... 111
付 録	改版履歴	... 112

CMOSデバイスの一般的注意事項

静電気対策（MOS全般）

注意 MOSデバイス取り扱いの際は静電気防止を心がけてください。

MOSデバイスは強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、NECが出荷梱包に使用している導電性のトレーやマガジン・ケース、または導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

未使用入力の処理（CMOS特有）

注意 CMOSデバイスの入力レベルは固定してください。

バイポーラやNMOSのデバイスと異なり、CMOSデバイスの入力に何も接続しない状態で動作させると、ノイズなどに起因する中間レベル入力が生じ、内部で貫通電流が流れて誤動作を引き起こす恐れがあります。プルアップかプルダウンによって入力レベルを固定してください。また、未使用端子が出力となる可能性（タイミングは規定しません）を考慮すると、個別に抵抗を介して V_{DD} またはGNDに接続することが有効です。

資料中に「未使用端子の処理」について記載のある製品については、その内容を守ってください。

初期化以前の状態（MOS全般）

注意 電源投入時、MOSデバイスの初期状態は不定です。

分子レベルのイオン注入量等で特性が決定するため、初期状態は製造工程の管理外です。電源投入時の端子の出力状態や入出力設定、レジスタ内容などは保証しておりません。ただし、リセット動作やモード設定で定義している項目については、これらの動作ののちに保証の対象となります。

リセット機能を持つデバイスの電源投入後は、まずリセット動作を実行してください。

OPENCAD は、日本電気株式会社の登録商標です。

Design Compiler は、日本 Synopsys 社の登録商標です。

Verilog-XL は、米国 Cadence Design Systems 社の商標です。

本製品が外国為替および外国貿易管理法の規定による規制貨物等（または役務）に該当するか否かは、ユーザ（仕様を決定した者）が判定してください。

- 本資料は、この製品の企画段階で作成していますので、予告なしに内容を変更することがあります。また本資料で扱う製品の製品化を中止することがあります。
 - 文書による当社の承諾なしに本資料の転載複製を禁じます。
 - 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
 - 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。
 - 当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意願います。
 - 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「特定水準」に分類しております。また、各品質水準は以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認の上ご使用願います。
 - 標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
 - 特別水準：輸送機器（自動車、列車、船舶等）、交通用信号機器、防災／防犯装置、各種安全装置、生命維持を直接の目的としない医療機器
 - 特定水準：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等
- 当社製品のデータ・シート／データ・ブック等の資料で、特に品質水準の表示がない場合は標準水準製品であることを表します。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は、必ず事前に当社販売窓口までご相談頂きますようお願い致します。

本版で改訂された主な箇所 (1/2)

箇 所	内 容
全般	NPB 周辺マクロの記述を削除
p.19	図 2 - 1 クロック制御回路の接続例 (ASIC 内部に発振回路を置いて振動子を外付けする場合) 修正
p.24	図 3 - 1 NB85E の ROM 領域設定 (1) 32 ビット×8K ワードのコンパイルド ROM を 1 個接続した場合 追加
p.25	図 3 - 2 ROM アクセス・タイミング 注を追加
p.26	3.2.1 ROM を 1 個接続する場合 追加
p.26	図 3 - 3 VFB へのコンパイルド ROM (32 ビット×8K ワード (1 個)) の接続例 追加
p.27-31	3.2.2 複数の ROM を接続する場合 追加
p.28	図 3 - 4 VFB へのコンパイルド ROM (32 ビット×4K ワード (2 個)) の接続例 修正
p.33	図 3 - 9 RAM アクセス・タイミング 修正
p.34	図 3 - 10 NB85E の RAM 領域設定 (2) 8 ビット×2K ワードのコンパイルド RAM を 8 個接続した場合 追加
p.35	図 3 - 11 VDB へのコンパイルド・メモリの接続例 (4 個接続した場合) 修正
p.36	図 3 - 12 VDB へのコンパイルド・メモリの接続例 (8 個接続した場合) 追加
p.37-39	3.3.1 RAM を 8 個接続した場合の動作 追加
p.41	図 4 - 1 VSB へのユーザ・ロジック接続例 修正
p.42	4.1.1 VSB 動作概要 修正
p.44	図 4 - 4 VSB タイミング (アドレス・ホールド) 修正
p.51	4.2 コンパイルド・メモリの接続 修正
p.56	第 5 章 メモリ・コントローラ (MEMC) の接続 説明を修正
p.57	図 5 - 1 NB85E, MEMC, 外部メモリ (SRAM, SDRAM) の接続例 修正
p.58	図 5 - 2 SRAM との接続例 修正
旧版 p.50-53	5.1 (2) レジスタ設定例, (3) 動作タイミング例 削除
旧版 p.55, 56	5.2 (2) レジスタ設定例, (3) 動作タイミング例 削除
p.60	5.3 SDRAM との接続 修正
p.61	図 5 - 4 SDRAM との接続例 修正
旧版 p.59-65	5.3 (2) レジスタ設定例, (3) 動作タイミング例 削除
p.63	6.2 (1) レジスタのマッピング 周辺 I/O 領域を修正
p.64	図 6 - 3 アドレス・デコーダの HDL 作成例 修正
p.65	図 6 - 4 ユーザ・ロジックの HDL 作成例 修正

本版で改訂された主な箇所 (2/2)

箇所	内容
p.67	図 6 - 6 リトライ機能付きユーザ・ロジックの HDL 作成例 修正
p.69	図 7 - 1 NB85E と命令キャッシュ (NB85E213) の接続例 修正
p.71	7.1.1 (7) プログラムの初期設定 追加
p.71	7.1.1 (8) NB85E の BHC レジスタの設定 追加
p.72	7.1.1 (9) テスト・バス自動結線ツールの対応 追加
p.72	7.1.1 (10) タグ・クリア手順 追加
p.73	図 7 - 3 NB85E とデータ・キャッシュ (NB85E263) の接続例 修正
p.74	7.2.1 (5) テスト・バス自動結線ツールの対応 追加
p.75	7.2.1 (6) その他 追加
p.75	7.2.1 (7) デバッグ中の動作 追加
p.79	図 8 - 4 インサーキット・エミュレータ接続回路例 (NB85E + RCU (NB85E901)) 修正
p.80	図 8 - 5 インサーキット・エミュレータ接続回路例 (NB85ET) 修正
p.81	8.2.4 N-Wire 型インサーキット・エミュレータとのタイミング設計 追加
p.86	図 9 - 2 スキュー, ホールド違反対策実施例 修正
p.90	9.5 (2) 製品仕様 修正
p.92	図 10 - 1 NB85E と周辺マクロの接続例 修正
p.94	10.2.1 (1) (b) 入力端子の処理 修正
p.94	10.2.1 (1) (c) 出力端子の処理 修正
p.95	10.2.1 (2) テスト・モード用端子 修正
p.95	10.2.1 (3) NB85E901 (RCU) 使用時の注意事項 修正
p.95	10.2.1 (4) NB85ET 使用時の注意事項 修正
p.98, 99	10.3 テスト・バス自動結線 追加
p.100-105	10.4 テスト・バス PINF ファイルの作成 / 修正方法 追加
p.106	第 11 章 (1) 接続チェック用テスト・パターンの作成 説明追加
p.112	付録 改版履歴 追加

本文欄外の 印は、本版で改訂された主な箇所を示しています。

巻末にアンケート・コーナを設けております。このドキュメントに対するご意見をお気軽にお寄せください。

はじめに

対象者 このマニュアルは、NEC の高速 / 高集積 CMOS CBIC「CB-9 ファミリ VX/VM タイプ」を使用して ASIC を設計するユーザを対象とします。

目的 このマニュアルは、32 ビット・マイクロプロセッサ・コア「NB85E, NB85ET」を内蔵した ASIC を設計するうえでの特有の設計方法や各種制限事項、注意事項などをユーザに理解していただくことを目的としています。

なお、この設計マニュアルに記載された事項（一般事項、注意事項、制限事項など）は必ずお守りください。お守りいただけない場合、ASIC の品質、性能の低下や動作の異常が生じることがあります。ただし、これらの記載事項は、回路を保証するものではなく、配置配線結果によっては、必要な機能を満足しない可能性があります。したがって、必ず動作検証を行ったうえで、開発を行ってください。

構成 このマニュアルは、大きく分けて次の内容で構成しています。

- クロック制御回路の接続
- キャッシュの接続
- VFB/VDB への ROM/RAM の接続
- インサーキット・エミュレータ（IE）との接続
- VSB への接続
- 注意事項
- メモリ・コントローラの接続
- テスト回路設計
- NPB への接続
- トータル・チップ・シミュレーション

読み方 このマニュアルの記載事項は NB85E と NB85ET に適用されますが、特に断りのないかぎり、NB85E をマイクロプロセッサ・コア（CPU コア）の代表として説明しています。NB85ET を使用する場合は、CPU コア名称と一部の端子名を次の表に従って読み替えてお使いください（端子機能は両製品で同一です）。

項目	NB85E を使用する場合 ^{注1} (このマニュアルで使用されている名称)	NB85ET を使用する場合 (下記の名称への読み替えが必要)
CPU コア	NB85E	NB85ET
端子名	DCRESZ ^{注2}	RESETZ
	DCSTOPZ	STOPZ
	DCNMI2-DCNMI0	NMI2-NMI0

注 1. NB85E901（ラン・コントロール・ユニット（RCU））が接続されているシステムを含みます。

注 2. NB85ET にも機能が異なる同一名称の端子があります。読み替える際には注意してください。

また、このマニュアルを使用する前には、必ず別冊の「CB-9 ファミリ VX/VM タイプ 設計マニュアル（A12745J）」をお読みください。

凡 例	データ表記の重み	: 左が上位桁, 右が下位桁
	アクティブ・ロウの表記	: xxxZ (端子, 信号名称のあとに Z), または xxxB (端子, 信号名称のあとに B)
	注	: 本文中につけた注の説明
	注意	: 気をつけて読んでいただきたい内容
	備考	: 本文の補足説明
	数の表記	: 2 進数 ...xxxx または xxxxB 10 進数...xxxx 16 進数...xxxxH

2 のべき数を示す接頭語 (アドレス空間, メモリ容量) :

K (キロ) ... $2^{10} = 1024$

M (メガ) ... $2^{20} = 1024^2$

G (ギガ) ... $2^{30} = 1024^3$

関連資料 関連資料は暫定版の場合がありますが, この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

- CB-9 ファミリ VX/VM タイプ 設計マニュアル (A12745J)
- CB-9 ファミリ VX/VM タイプ コア・ライブラリ 設計マニュアル
CPU コア, メモリ・コントローラ編 (A13195J)
- CB-9 ファミリ VX/VM タイプ 設計マニュアル
メモリ・マクロ (コンパイルド・タイプ) 編 (A12982J)
- NEC システム LSI 設計 OPENCAD[®] V5.4 ユーザーズ・マニュアル
OPC_VSHELL 編 (A15050J)
- NEC システム LSI 設計 OPENCAD V5.4 ユーザーズ・マニュアル
Verilog-XL[™] インタフェース編 (A15052J)
- NEC システム LSI 設計 OPENCAD V5.4 ユーザーズ・マニュアル
Design Compiler[®] インタフェース編 (A15058J)
- NB85E ユーザーズ・マニュアル ハードウェア編 (A13971J)
- NB85ET ユーザーズ・マニュアル ハードウェア編 (A14342J)
- メモリ・コントローラ ユーザーズ・マニュアル NB85E, NB85ET 編 (A14206J)
- 命令キャッシュ, データ・キャッシュ ユーザーズ・マニュアル NB85E, NB85ET 編 (A14247J)
- IE-V850E-MC, IE-V850E-MC-A ユーザーズ・マニュアル (U14487J)
- IE-V850E-MC-EM1-B, IE-V850E-MC-MM2 ユーザーズ・マニュアル (U14482J)

なお, 上記関連資料は予告なしに内容を変更することがあります。設計などには必ず最新の資料をご使用ください。

目 次

第 1 章 概 説 ... 16
1.1 NB85E 搭載時の CBIC 設計フロー ... 16
1.2 NB85E 搭載時に特有の項目 ... 17
1.3 システム例 ... 18
第 2 章 クロック制御回路の接続 ... 19
第 3 章 VFB/VDB への ROM/RAM の接続 ... 23
3.1 概 要 ... 23
3.2 VFB へのコンパイルド・メモリの接続 ... 23
3.2.1 ROM を 1 個接続する場合 ... 26
3.2.2 複数の ROM を接続する場合 ... 27
3.3 VDB へのコンパイルド・メモリの接続 ... 32
3.3.1 RAM を 8 個接続した場合の動作 ... 37
第 4 章 VSB への接続 ... 40
4.1 ユーザ・ロジックの接続 ... 40
4.1.1 VSB 動作概要 ... 42
4.1.2 回路例 ... 45
4.2 コンパイルド・メモリの接続 ... 51
第 5 章 メモリ・コントローラ (MEMC) の接続 ... 56
5.1 SRAM との接続 ... 58
5.2 ページ ROM との接続 ... 59
5.3 SDRAM との接続 ... 60
第 6 章 NPB への接続 ... 62
6.1 NPB の概要 ... 62
6.2 ユーザ・ロジックの接続 ... 63
第 7 章 キャッシュの接続 ... 69

7.1	命令キャッシュの接続 ...	69
7.1.1	命令キャッシュ使用時の注意事項 ...	70
7.2	データ・キャッシュの接続 ...	72
7.2.1	データ・キャッシュ使用時の注意事項 ...	74
第8章	インサーキット・エミュレータ (IE) との接続 ...	76
8.1	NB85E 用インサーキット・エミュレータ (IE-V850E-MC-A) を使用する場合 ...	76
8.2	N-Wire 型インサーキット・エミュレータ (IE-70000-MC-NW-A) を使用する場合 ...	77
8.2.1	インサーキット・エミュレータ接続コネクタ (ターゲット・システム側) ...	77
8.2.2	NB85E に RCU (NB85E901) を接続した場合の回路例 ...	79
8.2.3	NB85ET を接続した場合の回路例 ...	80
8.2.4	N-Wire 型インサーキット・エミュレータとのタイミング設計 ...	81
第9章	注意事項 ...	82
9.1	リード・サイクルからライト・サイクルに移行する際のバス・コンテンション (MEMC (NB85E500) 使用時) ...	82
9.2	Verilog シミュレーションに関する注意 ...	83
9.3	チップの BUNRI テスト時の注意 ...	84
9.4	タイミング調整 ...	85
9.4.1	クロック・スキュー, データ・ラインのホールド違反の調整方法 ...	85
9.4.2	デザイン・コンパイラを使用してホールド違反を解決する方法 ...	87
9.4.3	クロック・スキューの調整 ...	87
9.5	デバイス・ファイル ...	90
第10章	テスト回路設計 ...	91
10.1	テスト・モード時の周辺マクロ接続例 ...	92
10.2	テスト・モード時の各端子の処理 ...	93
10.2.1	NB85E, NB85ET の各端子の処理 ...	93
10.2.2	MEMC の各端子の処理 ...	96
10.3	テスト・バス自動結線 ...	98
10.3.1	テスト・バス自動結線方法 ...	98
10.3.2	ダミー・モデルによるテスト・バス接続確認 ...	99
10.4	テスト・バス PINF ファイルの作成 / 修正方法 ...	100
10.4.1	PINF ファイル作成例 ...	100
第11章	トータル・チップ・シミュレーション ...	106
11.1	接続チェック用テスト・パターンの作成 ...	107

11.2	マクロ間のタイミング検証 ...	110
第 12 章	ROM コードの作成 ...	111
付 録	改版履歴 ...	112

図の目次 (1/2)

図番号	タイトル, ページ
1 - 1	システム例 ... 18
2 - 1	クロック制御回路の接続例 (ASIC 内部に発振回路を置いて振動子を外付けする場合) ... 19
2 - 2	クロック制御回路のタイミング・チャート ... 20
2 - 3	クロック制御回路の HDL 作成例 ... 21
2 - 4	クロック制御回路の接続例 (PLL マクロを使用する場合) ... 22
3 - 1	NB85E の ROM 領域設定 ... 24
3 - 2	ROM アクセス・タイミング ... 25
3 - 3	VFB へのコンパイルド ROM (32 ビット×8K ワード (1 個)) の接続例 ... 26
3 - 4	VFB へのコンパイルド ROM (32 ビット×4K ワード (2 個)) の接続例 ... 28
3 - 5	ROM リード・タイミング (同一 MEM 領域の場合) ... 29
3 - 6	ROM リード・タイミング (MEM 領域が切り替わった場合) ... 30
3 - 7	CSB, OEB の生成部 ... 31
3 - 8	CSB, OEB のタイミング ... 31
3 - 9	RAM アクセス・タイミング ... 33
3 - 10	NB85E の RAM 領域設定 ... 34
3 - 11	VDB へのコンパイルド・メモリの接続例 (4 個接続した場合) ... 35
3 - 12	VDB へのコンパイルド・メモリの接続例 (8 個接続した場合) ... 36
3 - 13	RAM リード・タイミング (同一 RAM 領域からのリード) ... 37
3 - 14	RAM リード・タイミング (RAM 領域が切り替わった場合) ... 38
3 - 15	RAM ライト・タイミング ... 39
4 - 1	VSB へのユーザ・ロジック接続例 ... 41
4 - 2	VSB タイミング (ノー・ウエイト) ... 42
4 - 3	VSB タイミング (1 ウエイト) ... 43
4 - 4	VSB タイミング (アドレス・ホールド) ... 44
4 - 5	ノー・ウエイトで動作するユーザ・ロジックの HDL 作成例 ... 45
4 - 6	ノー・ウエイト時のタイミング・チャート ... 46
4 - 7	1 ウエイトで動作するユーザ・ロジックの HDL 作成例 ... 47
4 - 8	1 ウエイト挿入時のタイミング・チャート ... 48
4 - 9	アドレス・ホールド機能を使用したユーザ・ロジックの HDL 作成例 ... 49
4 - 10	アドレス・ホールド時のタイミング・チャート ... 50
4 - 11	コンパイルド・メモリの接続例 ... 52
4 - 12	コンパイルド・メモリ・アクセス時のタイミング・チャート ... 53
4 - 13	コンパイルド・メモリ・アクセス制御回路の HDL 作成例 ... 54
5 - 1	NB85E, MEMC, 外部メモリ (SRAM, SDRAM) の接続例 ... 57
5 - 2	SRAM との接続例 ... 58

図の目次 (2/2)

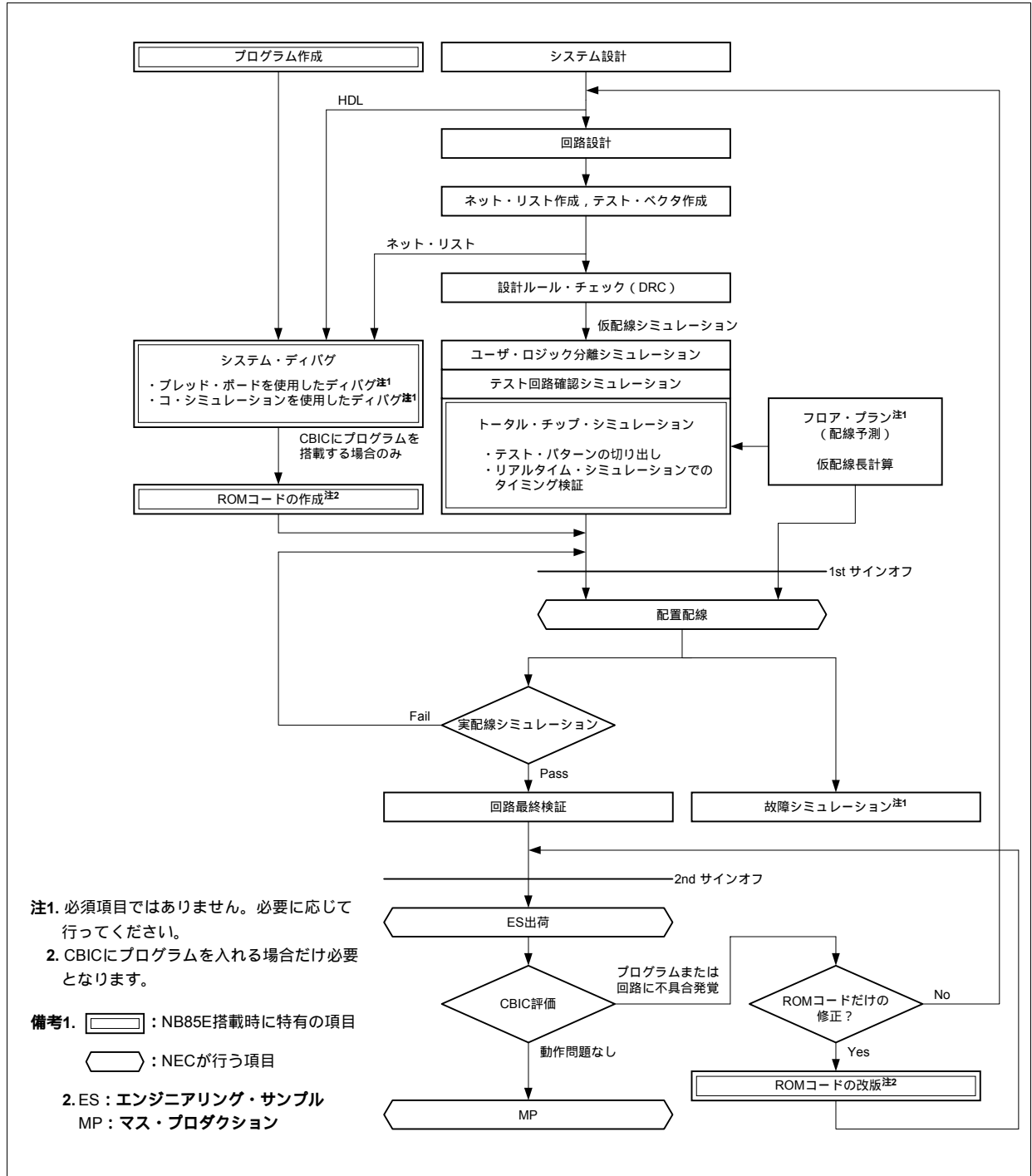
図番号	タイトル, ページ
5 - 3	ページ ROM との接続例 ... 59
5 - 4	SDRAM との接続例 ... 61
6 - 1	NPB 動作タイミング・チャート ... 62
6 - 2	ユーザ・ロジックの接続例 ... 64
6 - 3	アドレス・デコーダの HDL 作成例 ... 64
6 - 4	ユーザ・ロジックの HDL 作成例 ... 65
6 - 5	ユーザ・ロジックの動作タイミング・チャート ... 66
6 - 6	リトライ機能付きユーザ・ロジックの HDL 作成例 ... 67
6 - 7	リトライ機能付きユーザ・ロジックの動作タイミング・チャート ... 68
7 - 1	NB85E と命令キャッシュ (NB85E213) の接続例 ... 69
7 - 2	キャッシュ領域設定例 ... 71
7 - 3	NB85E とデータ・キャッシュ (NB85E263) の接続例 ... 73
8 - 1	NB85E 用インサーキット・エミュレータ (IE-V850E-MC-A) の接続 ... 76
8 - 2	N-Wire 型インサーキット・エミュレータの接続 ... 77
8 - 3	インサーキット・エミュレータ接続コネクタ (ターゲット・システム側) のピン配置図 ... 77
8 - 4	インサーキット・エミュレータ接続回路例 (NB85E + RCU (NB85E901)) ... 79
8 - 5	インサーキット・エミュレータ接続回路例 (NB85ET) ... 80
8 - 6	NB85ET と N-Wire 型インサーキット・エミュレータ間の遅延 ... 81
9 - 1	ユーザ・ロジック設計例 ... 84
9 - 2	スキュー, ホールド違反対策実施例 ... 86
10 - 1	NB85E と周辺マクロの接続例 ... 92
10 - 2	ユーザ・ロジック設計例 ... 94
10 - 3	ユーザ・ロジック設計例 ... 96
10 - 4	PINF ファイル作成例 (NB85E + NU85E500 + NU85E502 + NB85E901 + NB85E212) ... 101
10 - 5	PINF ファイル作成例 (NB85ET + NU85E500 + NU85E502 + NB85E212) ... 104
11 - 1	トータル・チップ・シミュレーション用ブロック図 (VFB に ROM がいない場合) ... 107
11 - 2	トータル・チップ・シミュレーション用ブロック図 (VFB に ROM があり, VSB に仮想 ROM を接続してテスト・パターンを作成する場合) ... 108
11 - 3	トータル・チップ・シミュレーション用ブロック図 (VFB に ROM があり, その ROM にテ スト用プログラムを書き込む場合) ... 109

表の目次

表番号	タイトル, ページ
4 - 1	ユーザ・ロジックのインタフェース信号 ... 40
6 - 1	ユーザ・ロジックのアドレス割り当て例 ... 63
8 - 1	インサートキット・エミュレータ接続コネクタ (ターゲット・システム側) ピン機能 ... 78

第1章 概 説

1.1 NB85E 搭載時の CBIC 設計フロー



1.2 NB85E 搭載時に特有の項目

(1) トータル・チップ・シミュレーション

トータル・チップ・シミュレーションは、NB85E を動作させてマクロ間の接続チェックを行うシミュレーションです。

シミュレーション方法、注意事項などについては第11章 トータル・チップ・シミュレーションを参照してください。

(2) システム・ディバグ

開発する CBIC を含むターゲット・システム全体のディバグを行います。

(a) ブレッド・ボードを使用したディバグ

<1> NB85E 用インサーキット・エミュレータ (IE-V850E-MC-A) を使用したディバグ

汎用 LSI, FPGA, G/A を使用したブレッド・ボードを作成し、ターゲット・システムのディバグを行います。

IE-V850E-MC-A との接続については、8.1 NB85E 用インサーキット・エミュレータ (IE-V850E-MC-A) を使用する場合は参照してください。

<2> N-Wire 型インサーキット・エミュレータ (IE-70000-MC-NW-A) を使用したディバグ

NB85ET は、オンチップ・ディバグ機能をサポートするディバグ・コントロール・ユニット (DCU) を内蔵しているため、JTAG をベースとする専用ディバグ・インタフェースを介して、IE-70000-MC-NW-A と接続することにより、ES を用いたリアルタイム・ディバグが可能です。

また、NB85E の場合は、NB85E901 (ラン・コントロール・ユニット (RCU)) を接続すれば同様のディバグを行うことができます。

ディバグ	NB85ET	NB85E + NB85E901
ブレーク		
イベント検出		×
トレース		×

インサーキット・エミュレータとの接続については、8.2 N-Wire 型インサーキット・エミュレータ (IE-70000-MC-NW-A) を使用する場合は参照してください。

(b) コ・シミュレーションを使用したディバグ

コ・シミュレーションを使用して、ターゲット・システムのディバグを行います。

本来、ES 完成後でなければ、インサーキット・エミュレータによるハードウェア/ソフトウェアの統合ディバグはできませんが、コ・シミュレーションでは、ES 完成前のシステム設計段階で、ハードウェア/ソフトウェア両方のターゲットレス・ディバグが可能になります。

詳細については、NEC にお問い合わせください。

(3) ROM コードの作成, 改版

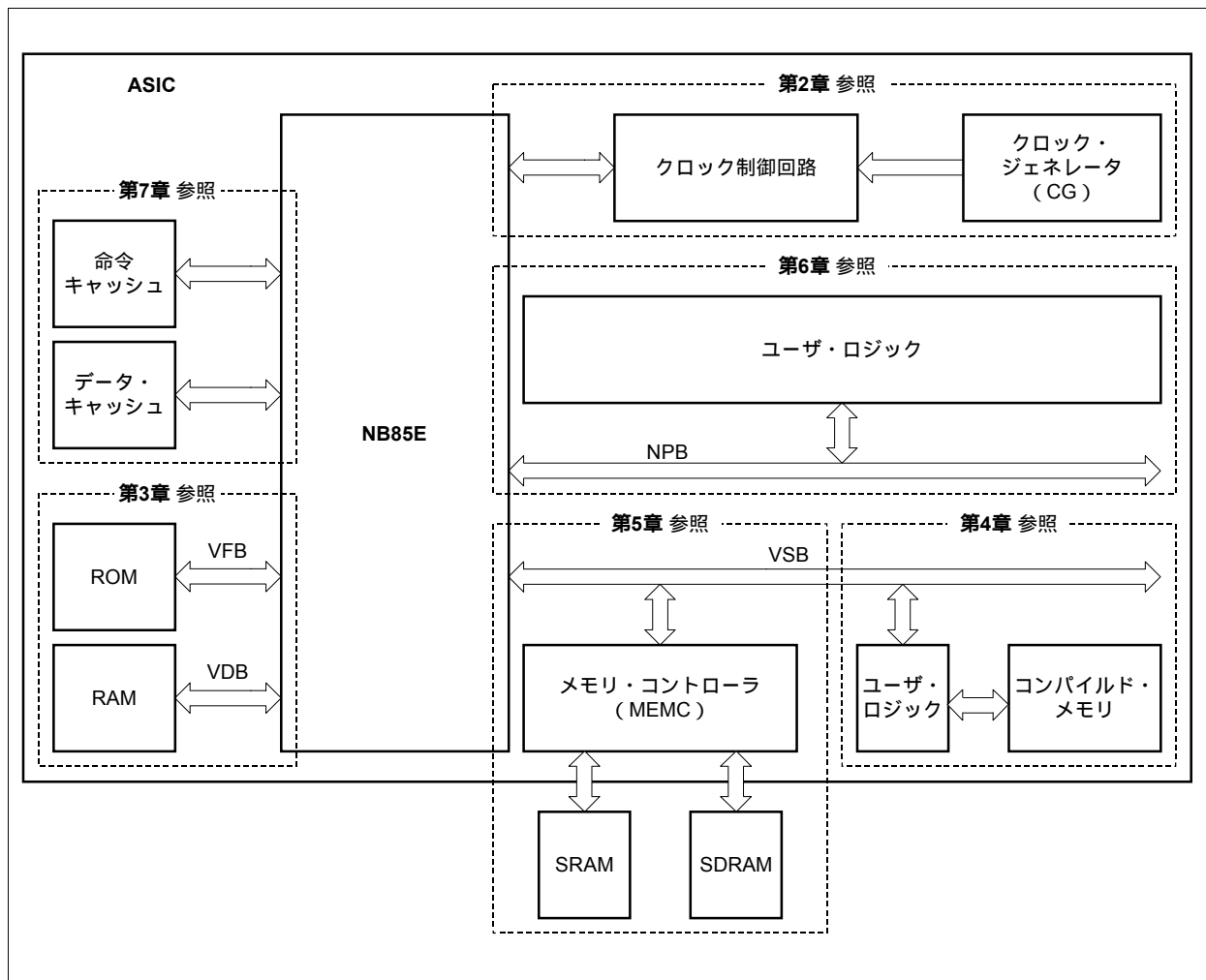
NEC へのサインオフの際は, 指定のフォーマットで行っていただく必要があります。

フォーマットに関しては, 第12章 ROM コードの作成を参照してください。

1.3 システム例

このマニュアルでは, 次に示す ASIC を例として, NB85E と各周辺マクロの接続方法を中心に説明します。

図1-1 システム例

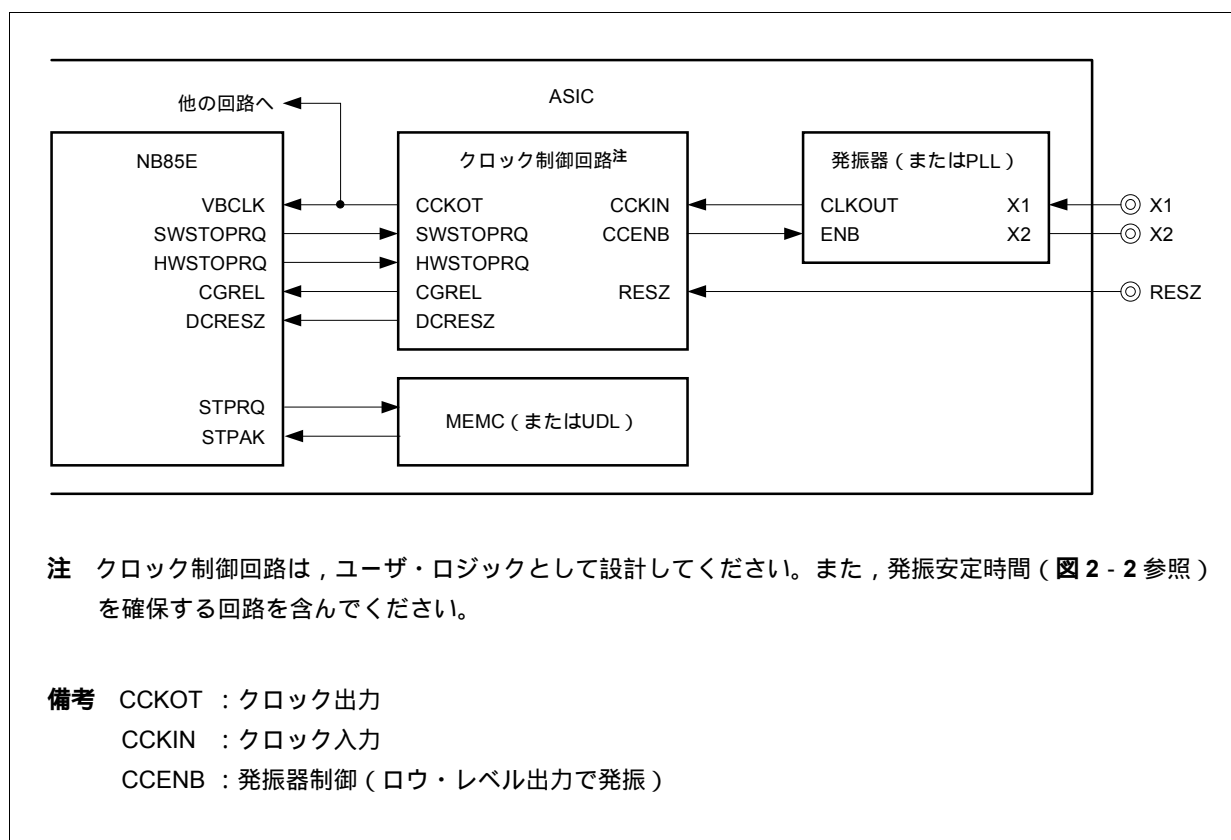


第2章 クロック制御回路の接続

NB85E やほかの回路へのクロック制御については、お客様で設計していただくことになります。

NB85E のソフトウェア/ハードウェア STOP モードを使用する場合、発振器（または、外部入力クロック）と NB85E の間や、発振器（または、外部入力クロック）と VBCLK が関係する他の回路の間には、クロック制御回路を接続して発振安定時間が十分確保でき、かつ NB85E に安定したクロックを供給できるようにしてください。

図2-1 クロック制御回路の接続例（ASIC内部に発振回路を置いて振動子を外付けする場合）



クロック制御回路の CCENB 端子からロウ・レベルが出力されると発振器が発振します。発振安定時間のカウンタは、クロック制御回路内部のカウンタ・レジスタ（CNTR）が行います。CNTR の目安値は次のような関係式で決まります。

$$\text{CNTR} = \text{発振安定時間 [s]} \times \text{クロック周波数 [Hz]}$$

例 出力クロック周波数が 20MHz で発振安定時間が 1ms の発振器を使用する場合

$$\text{CNTR} = 1 \times 10^{-3} \times 20 \times 10^6 = 20000 \text{ (4E20H)}$$

注意 NB85E の STPAK 端子が未使用端子の場合、STPAK 端子にはハイ・レベルを入力してください。ハイ・レベルを入力しないと、HWSTOPRQ 信号と SWSTOPRQ 信号がアクティブにならないため、STOP モードに移行できなくなります。

図2-1のクロック制御回路の各信号のタイミング・チャートを図2-2に、HDL作成例を図2-3に示します。

図2-2 クロック制御回路のタイミング・チャート

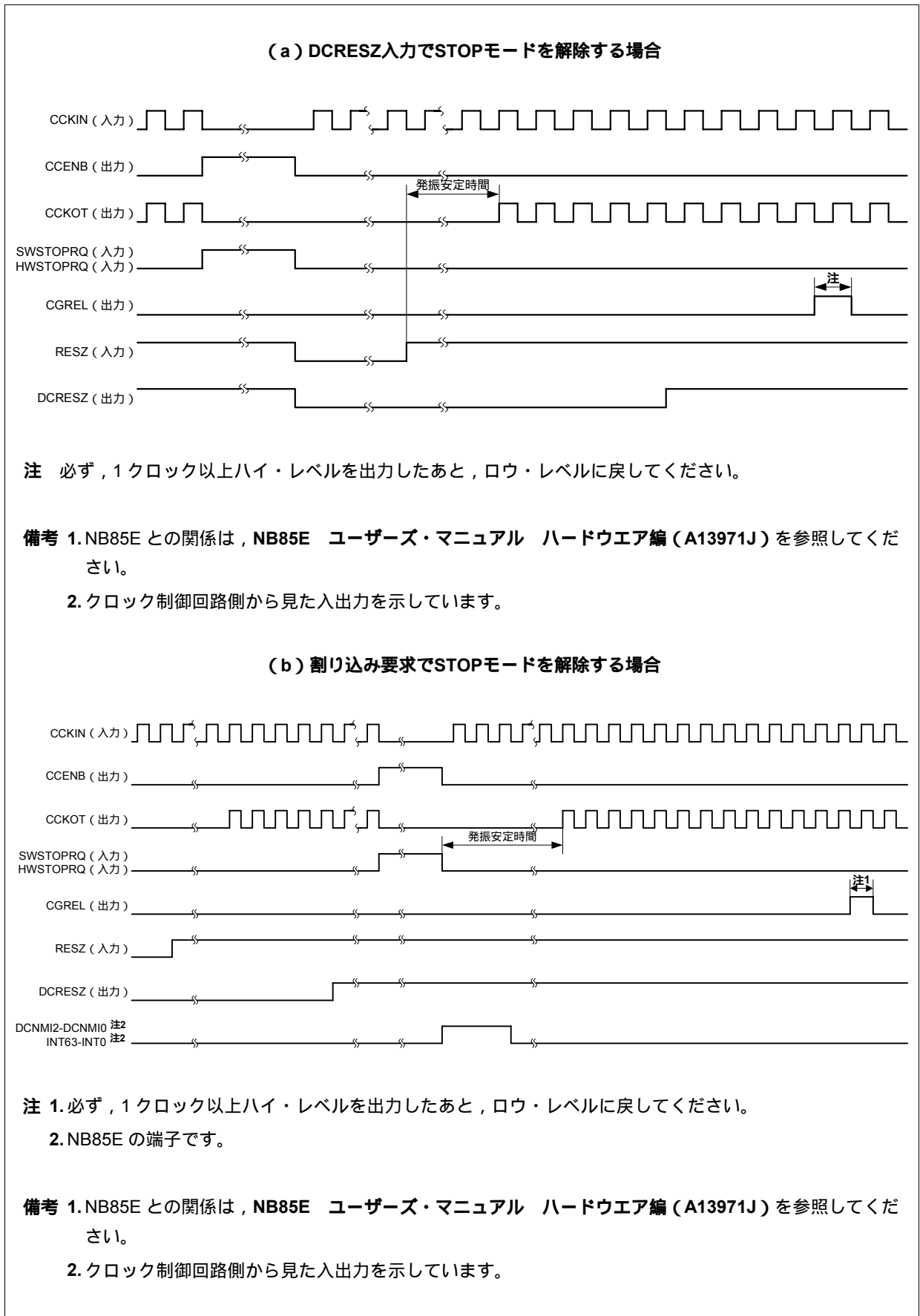


図2-3の例では、NB85Eのリセット入力(DCRESZ)も制御しています。これは、リセット期間中に安定したクロックを確実に供給するためのものです。また、DCRESZ信号はRESZ信号(システム・リセット)が立ち上がったあと、発振安定時間を確保してから立ち上がるようになっています。

図2-3 クロック制御回路のHDL作成例

```

module CLKCTL(
    CCKOT, SWSTOPRQ, HWSTOPRQ, CGREL,
    CCKIN, CCENB, DCRESZ, RESZ
);

output    CCKOT;           // NB85E の VBCLK 端子への出力
input     SWSTOPRQ;       // NB85E の SWSTOPRQ 端子からの入力
input     HWSTOPRQ;       // NB85E の HWSTOPRQ 端子からの入力
output    CGREL;          // NB85E の CGREL 端子への出力
input     CCKIN;          // 発振器 (または PLL) の CLKOUT 端子からの入力
output    CCENB;          // 発振器 (または PLL) の ENB 端子への出力
output    DCRESZ;         // NB85E の DCRESZ 端子への出力
input     RESZ;           // システム・リセット入力

reg [7:0] CNTR;           // 発振安定時間カウンタ。ビット幅は発振安定時間に合わせて調整。
reg       CKOTEN;         // クロック出力許可
reg       DCRESZ;         // NB85E リセット

wire      STOPRQZ = ~(SWSTOPRQ | HWSTOPRQ);

wire      CGREL = ( CNTR == 8'hFE ); // CGREL 出力タイミング設定
wire      CCKOT = CKOTEN & CCKIN;
wire      CCENB = ~STOPRQZ;

// synopsys async_set_reset "DCRESZ"
always @( negedge CCKIN or negedge STOPRQZ or negedge RESZ ) begin
    if ( ~RESZ ) begin
        CNTR        <= 8'h00;
        CKOTEN      <= 1'b0;
        DCRESZ      <= 1'b0;
    end
    else begin
        if( ~STOPRQZ ) begin
            CNTR        <= 8'h00;
            CKOTEN      <= 1'b0;
        end
        else begin
            if ( CNTR == 8'hF0 ) CKOTEN    <= 1'b1; // クロック出力タイミング設定。発振安定時間は F0H。
            if ( CNTR == 8'hF5 ) DCRESZ    <= 1'b1; // DCRESZ 出力タイミング設定
            if ( CNTR != 8'hFF ) CNTR      <= CNTR + 1; // 内部カウンタ停止タイミング設定
        end
    end
end

endmodule

```

なお，PLL マクロを使用する場合は，図 2 - 4 に示すような構成になります。

図2 - 4 クロック制御回路の接続例（PLLマクロを使用する場合）

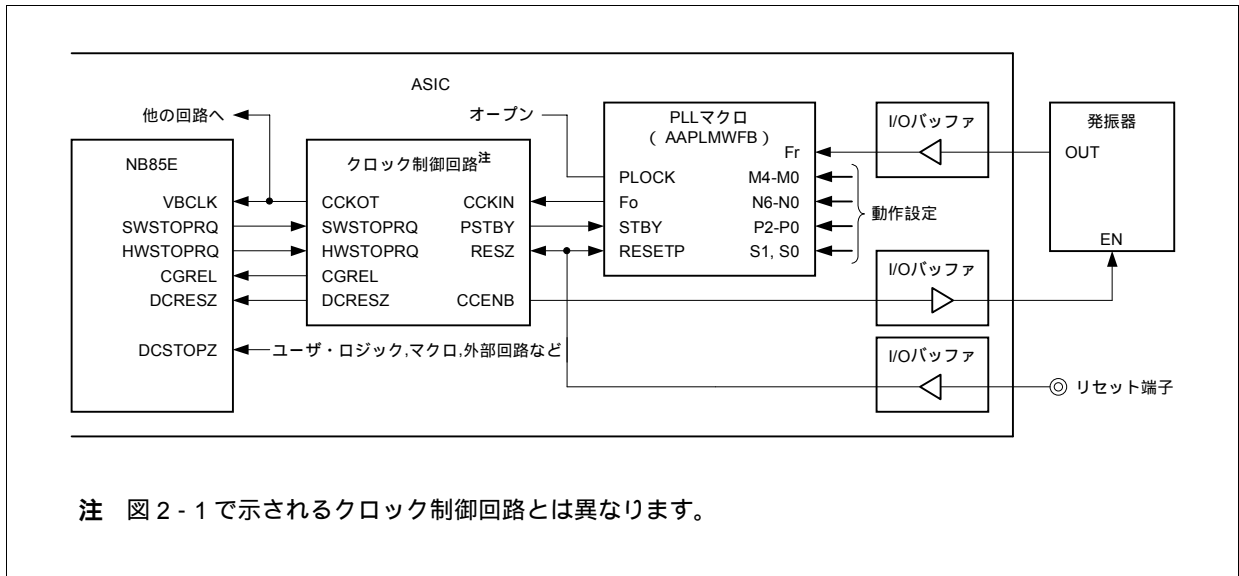


図 2 - 4 の PLL マクロとクロック制御回路について次に説明します。

(1) PLL マクロ

「AAPLMWFB」を例としています（マクロの詳細については，NEC にお問い合わせください）。入力クロックは，5MHz-160MHz，出力クロックは，50MHz-250MHz に対応しています。

図 2 - 4 では，各カウンタ値の設定を行う M4-M0, N6-N0, P2-P0 端子と VCO レンジ設定切り替えの設定を行う S1, S0 の各端子は，次のような設定になっており，Fr 端子に 16.384MHz のクロックを入力すると Fo 端子から 49.152MHz（50%デューティ）の信号が得られます。

- M4-M0 : L, L, L, L, H
- N6-N0 : L, L, L, L, L, H, H
- P2-P0 : L, L, H
- S1, S0 : L, L

備考 L: ロウ・レベル入力
H: ハイ・レベル入力

(2) クロック制御回路

図 2 - 1 に示す制御回路での STOP モード時に PLL をスタンバイ・モードに設定するための制御端子（PSTBY）を追加する必要があります。

また，発振安定時間は，PLL（ tw_{PLL} ），OSC（ tw_{OSC} ）の両方を発振させる場合，「 $tw_{PLL}+tw_{OSC}$ 」の時間を考慮する必要があります。

第3章 VFB/VDB への ROM/RAM の接続

3.1 概 要

NB85E は、ROM の直結が可能な V850E フェッチ・バス (VFB) と RAM の直結が可能な V850E データ・バス (VDB) を備えています。

VFB は命令フェッチ用に、VDB はデータ・アクセス用に最適化されたバスです。

これらのバスは、VSB から独立しているため、VSB が他のマクロに占有されていても命令フェッチまたは、データ・アクセスが行えます。

3.2 VFB へのコンパイルド・メモリの接続

VFB は、データ・アクセスに最適化された 32 ビット・アクセス固定の命令フェッチ用バスで、1 クロック・アクセスが可能です。

IFIROME 端子にハイ・レベルを入力すると VFB に接続した ROM から命令フェッチができるようになります。

なお、CB-9 ファミリ VX/VM タイプで使用できるコンパイルド ROM は、高速同期式 ROM です。

VFB にコンパイルド ROM を接続する場合、CPU のクロック周波数と ROM のアクセス・タイム、サイクル・タイムの関係を計算して、十分にタイミングが合うことを検討しておく必要があります (9.4 タイミング調整参照)。

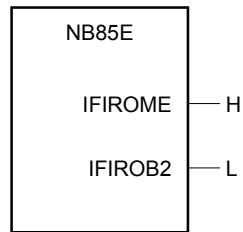
図 3 - 1 に NB85E の ROM 領域設定、図 3 - 2 に ROM アクセス・タイミング、図 3 - 3 に CB-9 ファミリ VX タイプ用 32 ビット×8K ワードのコンパイルド ROM を 1 個接続した例、図 3 - 4 に CB-9 ファミリ VX タイプ用 32 ビット×4K ワードのコンパイルド ROM を 2 個接続した例を示します。

- 備考 1.**コンパイルド・メモリの詳細については、**CB-9 ファミリ VX/VM タイプ 設計マニュアル メモリ・マクロ (コンパイルド・タイプ) 編 (A12982J)** を参照してください。
- 2.**コンパイルド・メモリの作成方法は、**NEC システム LSI 設計 OPENCAD V5.4 ユーザーズ・マニュアル OPC_VSHELL 編 (A15050J)** を参照してください。
- 3.**コンパイルド・メモリを使用する場合は、ASIC 標準のテスト手法を使用するため、テスト配線が必要になります。テスト手法については、**CB-9 ファミリ VX/VM タイプ 設計マニュアル (A12745J)** を参照してください。

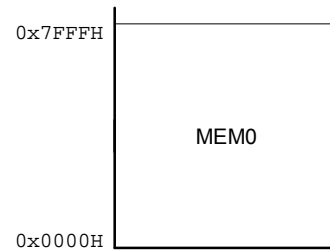
図3 - 1 NB85EのROM領域設定

(1) 32ビット×8KワードのコンパイルドROMを1個接続した場合

(a) 端子設定



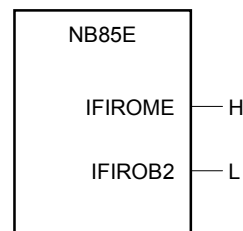
(b) メモリ・マップ



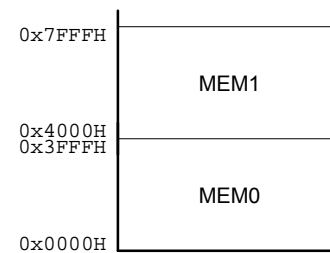
備考 L : ロウ・レベルを入力
H : ハイ・レベルを入力

(2) 32ビット×4KワードのコンパイルドROMを2個接続した場合

(a) 端子設定

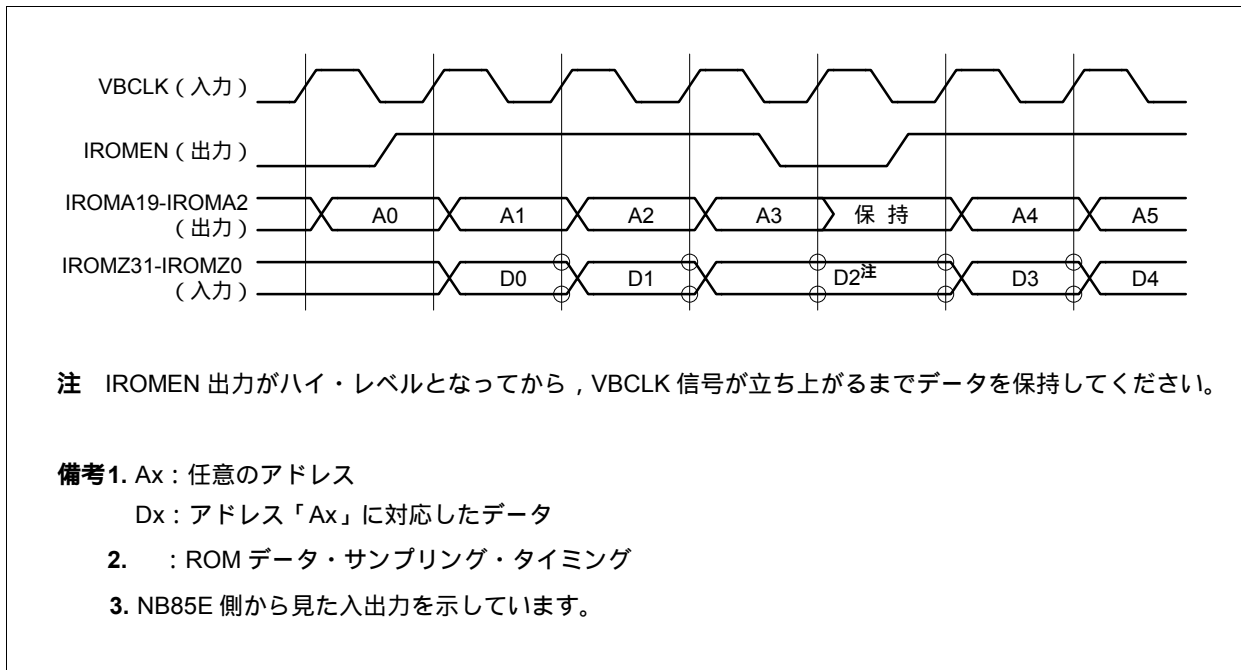


(b) メモリ・マップ



備考 L : ロウ・レベルを入力
H : ハイ・レベルを入力

図3 - 2 ROMアクセス・タイミング



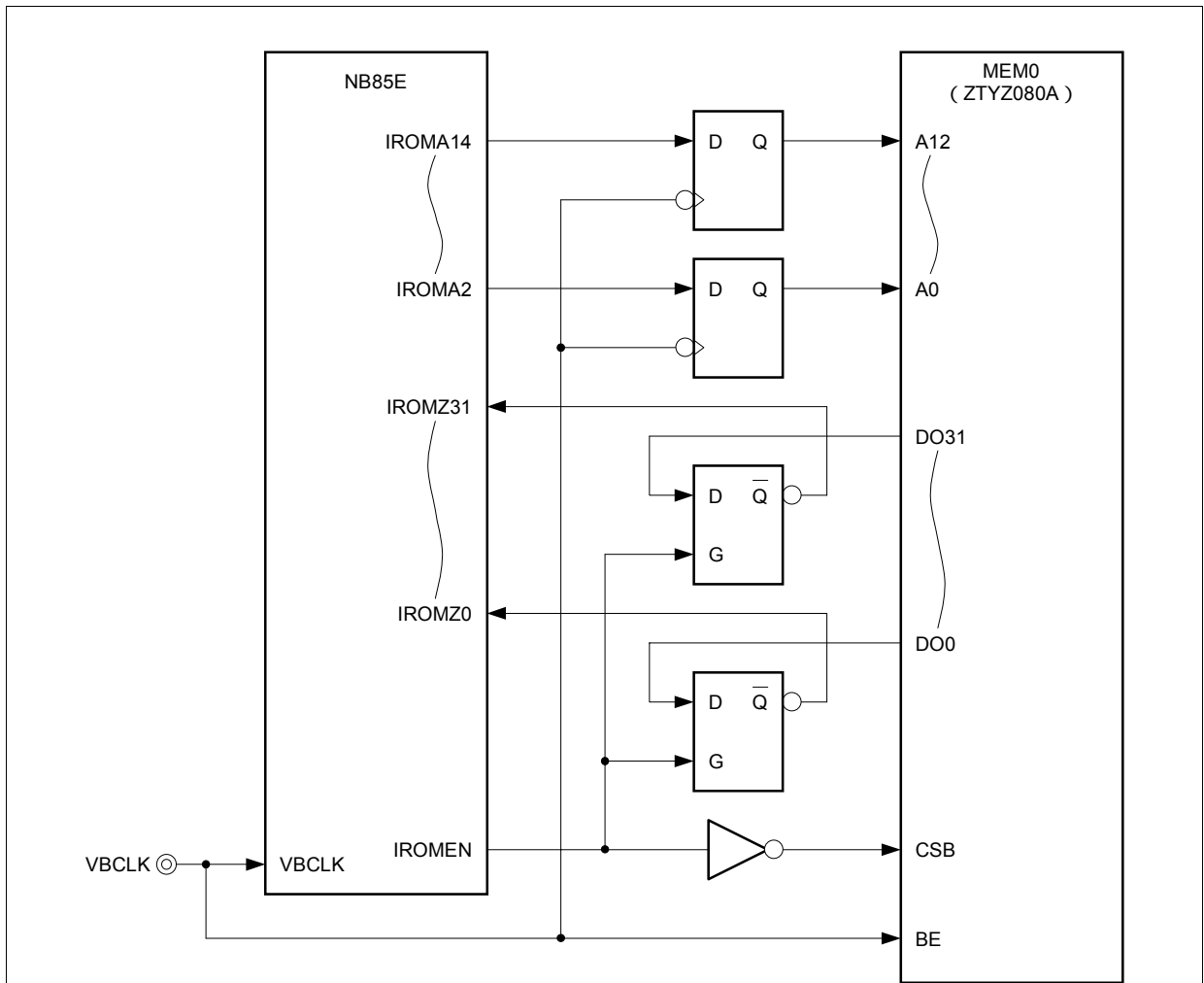
3.2.1 ROMを1個接続する場合

図3-3にCB-9ファミリVXタイプ用32ビット×8KワードのコンパイルドROMを接続した例を示します。

NB85EのVFBは、任意のアドレスとそのアドレスに対するデータが1サイクルだけシフトする「パイプライン方式」の動作をします。ここでは、アドレスのセットアップ/ホールド時間を確保しやすくするためにNB85EのIROMA14-IROMA2端子とコンパイルドROMのA12-A0端子の間にF/Fを挿入しています。また、IROMENがロウ・レベルの間データを保持するために、コンパイルドROMのDO31-DO0端子とNB85EのIROMZ31-IROMZ0端子の間にレベル・ラッチを挿入しています。

コンパイルドROMのアドレスの割り当ては、0x00000H-0x07FFFH番地に行っています。

図3-3 VFBへのコンパイルドROM (32ビット×8Kワード (1個)) の接続例



- 備考 1.** データ・バスは、実際のデータに対して論理が反転しているため、シミュレーションでデータをダンプする場合は注意してください。
- 2.** 図はノーマル端子の配線方法を示しています。テスト用端子は、ASICのテスト手法に準拠して配線してください（詳細については、**CB-9ファミリVX/VMタイプ 設計マニュアル (A12745J)**を参照してください）。

3.2.2 複数の ROM を接続する場合

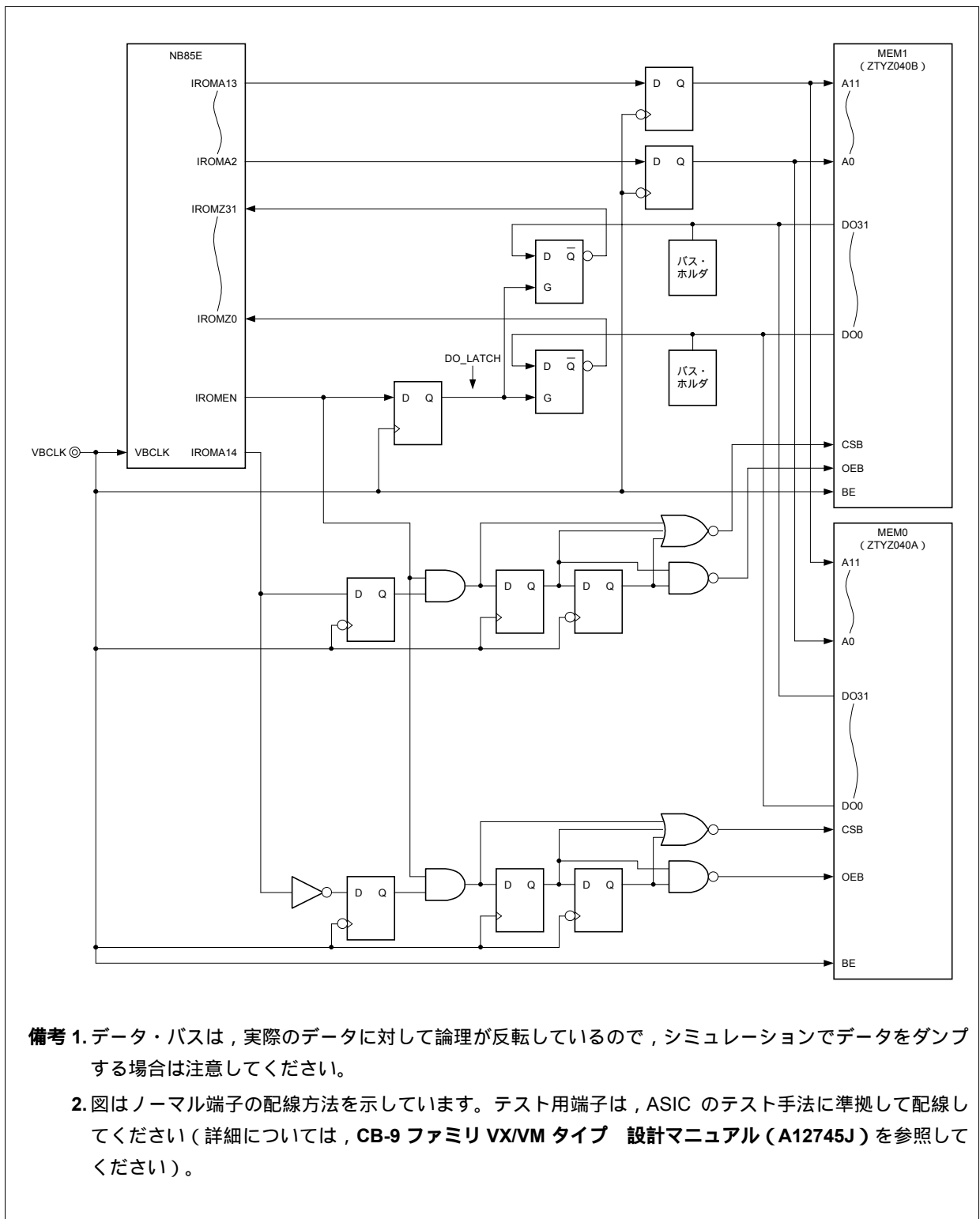
図 3 - 4 に CB-9 ファミリ VX タイプ用 32 ビット×4K ワードのコンパイルド ROM を 2 個接続した場合の接続例を示します。ROM 領域のメモリ・マップについては図 3 - 1 (2) 32 ビット×4K ワードのコンパイルド ROM を 2 個接続した場合を参照してください。

複数の ROM (たとえば MEM0 と MEM1) を接続する場合、MEM0 と MEM1 にまたがる領域の連続フェッチや、MEM0、MEM1 間の分岐時に ROM からデータをリードし終える前に CSB がインアクティブ・レベルに変化するため、正常にリードできない現象が起こり得ます。

これを回避するために、図 3 - 4 の接続例では NB85E の IROMEN 端子とコンパイルド ROM の CSB 端子の間に、領域切り替わりでの CSB の変化タイミングを調整する回路を挿入しています。また、この回路は MEM0、MEM1 のアドレス・デコーダも兼用しています。

コンパイルド ROM のアドレスの割り当ては、MEM0 を 0x0000H-0x3FFFH 番地に、MEM1 を 0x4000H-0x7FFFH 番地に行っています。

図3-4 VFBへのコンパイルROM (32ビット×4Kワード(2個))の接続例



- 備考 1.** データ・バスは、実際のデータに対して論理が反転しているため、シミュレーションでデータをダンプする場合は注意してください。
- 2.** 図はノーマル端子の配線方法を示しています。テスト用端子は、ASIC のテスト手法に準拠して配線してください（詳細については、**CB-9 ファミリ VX/VM タイプ 設計マニュアル (A12745J)** を参照してください）。

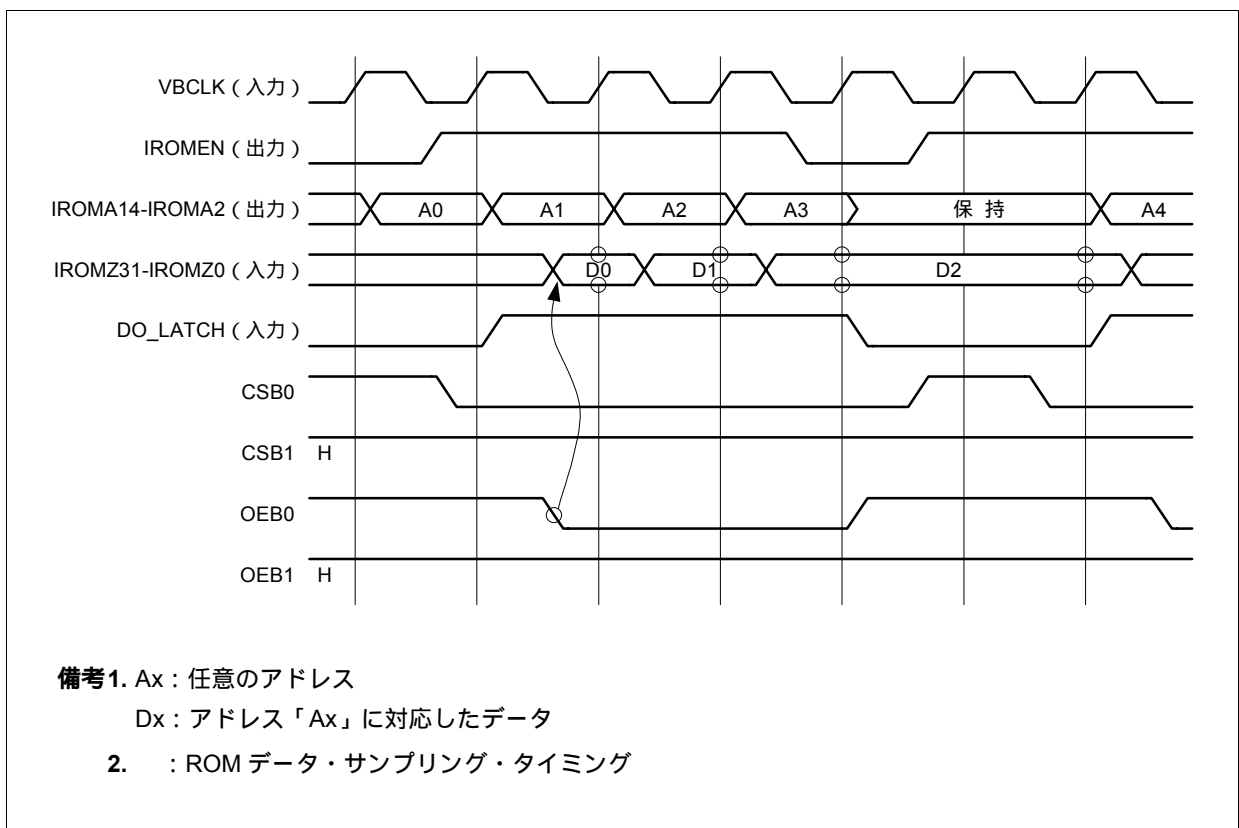
(1) 同一 MEM 領域からの ROM リード

アドレスをデコードした結果が MEM0 領域である状態 (図 3 - 4 では IROMA14 がロウ・レベル) で、IROMEN がアクティブになるとすぐに CSB0 がアクティブになります。CSB0 がアクティブになった次のクロックの立ち上がりで MEM0 側でアドレスがラッチされ、次のクロックの立ち下がりで OEB0 がアクティブになり、有効なデータが IROMZ31-IROMZ0 に入力されます。

IROMEN がインアクティブになった次の立ち上がりで、OEB0 がインアクティブとなると同時に IROMZ31-IROMZ0 はそのときの値を保持します。これは D0 と IROMZ31-IROMZ0 の間のレベル・ラッチによって、IROMEN がインアクティブである間、ROM に対してリードを行わずに値を保持するためです。OEB0 がインアクティブとなった次のクロックの立ち下がりで、CSB0 はインアクティブになります。

図 3 - 5 に MEM0 領域の連続アクセスした場合のタイミングを示します。

図3 - 5 ROMリード・タイミング (同一MEM領域の場合)



(2) MEM領域が切り替わった場合のROMリード

図3-6にA0-A2がMEM0領域、A3-A6がMEM1領域で、その間を連続アクセスしたときのCSB0, CSB1, OEB0, OEB1のタイミングを示します。

IROMA14-IROMA2がA3になり、ROMのアドレスがMEM0領域からMEM1領域に切り替わっても、A2に対するデータ(D2)を読み込むタイミングはさらに1クロック後のVBCLKの立ち上がりであるため、OEB0を1クロック間、CSB0を1.5クロック伸ばしています。CSB1はD3を読み出す準備のため、A3のクロック立ち下がりでアクティブにしています。OEB1は、MEM0とMEM1のデータの衝突を防ぐため、OEB0がインアクティブになってから0.5クロック後にアクティブにしています。

図3-6 ROMリード・タイミング (MEM領域が切り替わった場合)

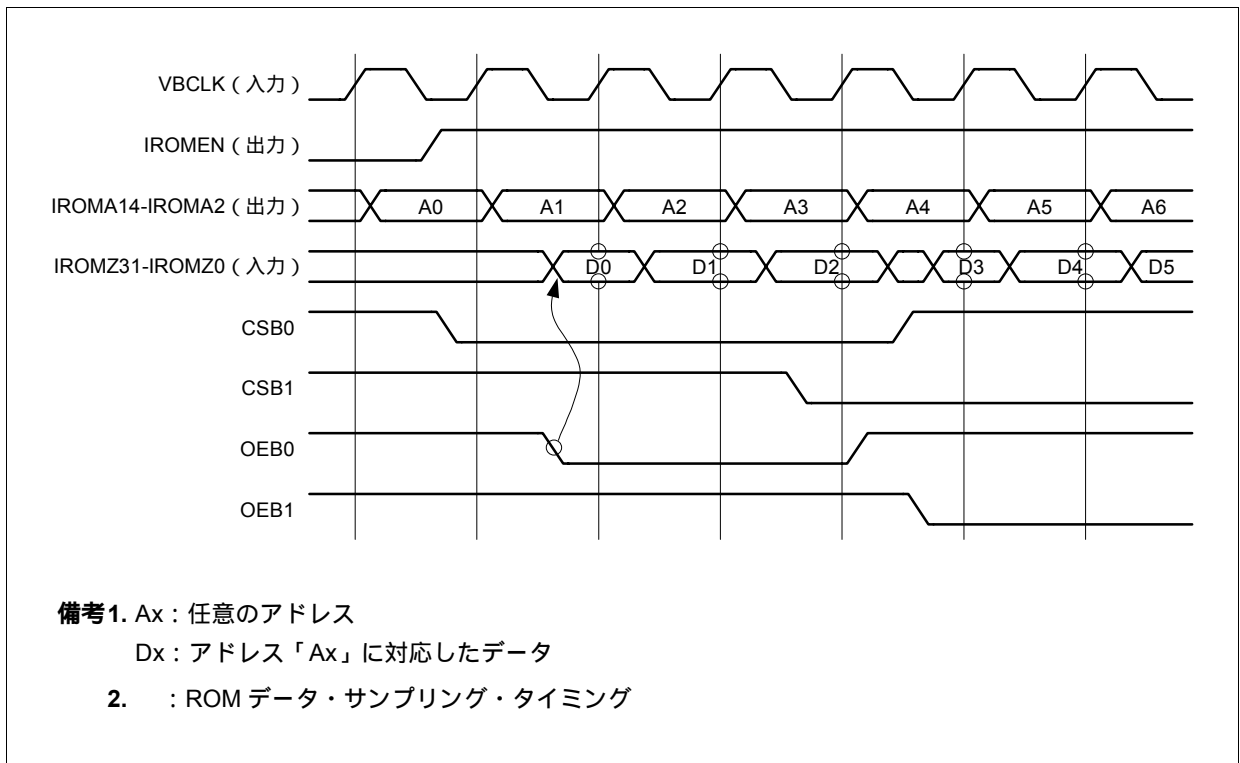


図3-7にCSB0, OEB0の生成部, 図3-8にCSB0, OEB0のタイミングを示します(図3-7の<a>-<c>は, 図3-8のMEM0側の<a>-<c>に対応しています)。

図3-7 CSB, OEBの生成部

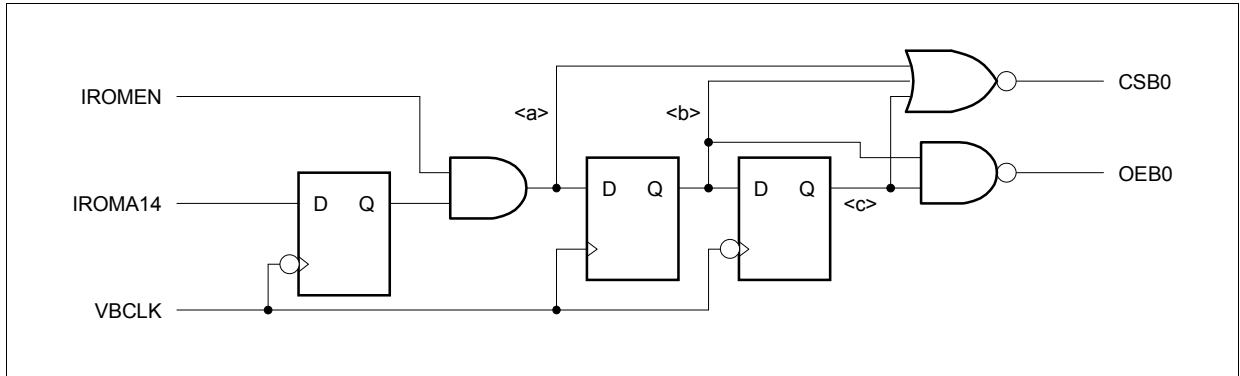
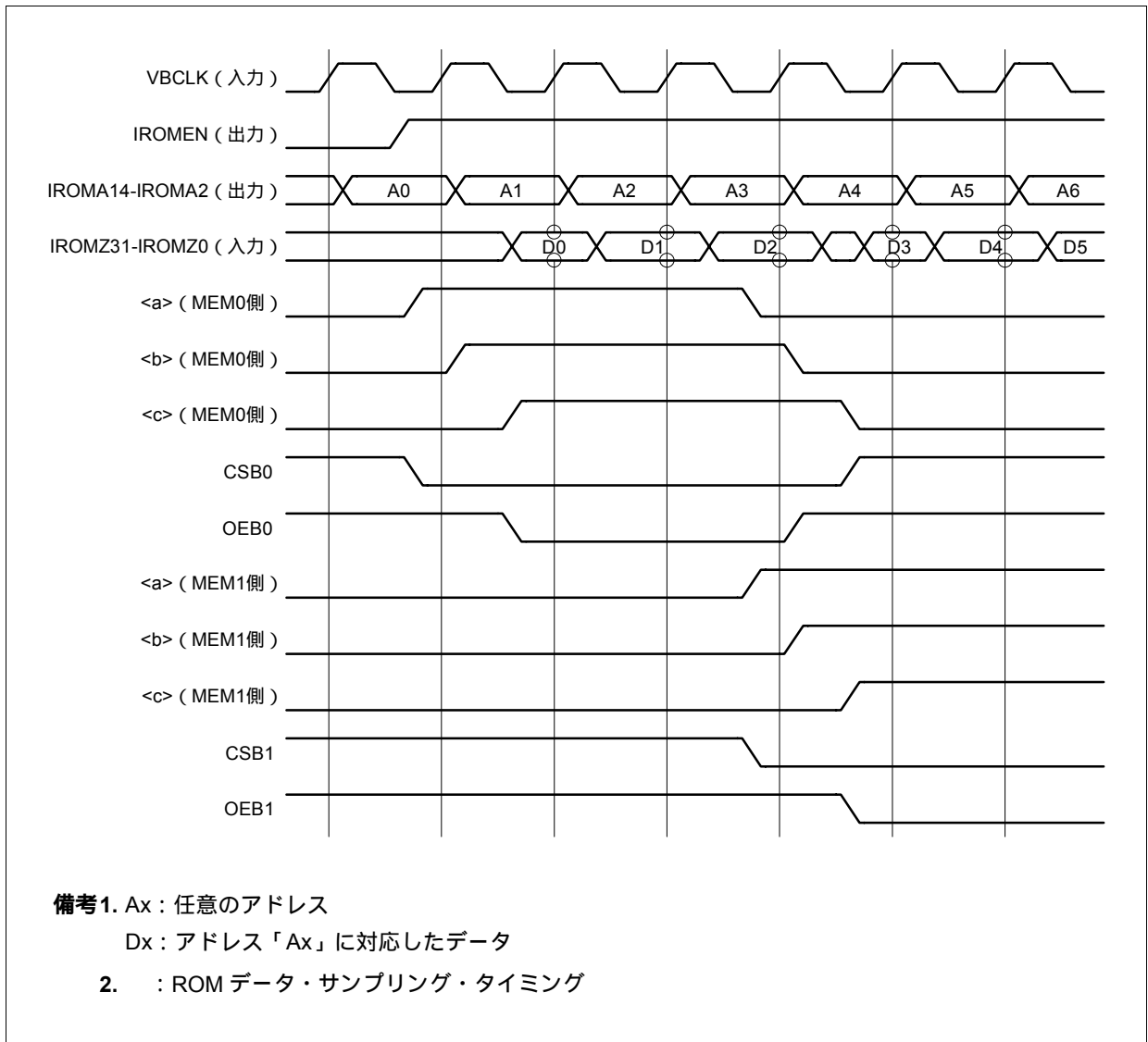


図3-8 CSB, OEBのタイミング



3.3 VDB へのコンパイルド・メモリの接続

VDB は、データ・アクセスに最適化されたバスで、最高 1 クロック・アクセスが可能です。

8 ビット、16 ビット、32 ビットの各単位でアクセスできます。

VDB にコンパイルド RAM を接続する場合、CPU のクロック周波数とコンパイルド RAM のアクセス・タイム、サイクル・タイムの関係を計算して、十分にタイミングが合うことを検討しておく必要があります（9.4 タイミング調整参照）。

図 3 - 9 に RAM アクセス・タイミング、図 3 - 10 に NB85E の RAM 領域設定、図 3 - 11 に CB-9 ファミリ VX タイプ用 8 ビット×2K ワードのコンパイルド RAM を 4 個接続した例、図 3 - 12 に 8 ビット×2K ワードのコンパイルド RAM を 8 個接続した例を示します。

- 備考** 1.コンパイルド・メモリの詳細については、**CB-9 ファミリ VX/VM タイプ 設計マニュアル メモリ・マクロ (コンパイルド・タイプ) 編 (A12982J)** を参照してください。
- 2.コンパイルド・メモリの作成方法は、**NEC システム LSI 設計 OPENCAD V5.4 ユーザーズ・マニュアル OPC_VSHELL 編 (A15050J)** を参照してください。
- 3.コンパイルド・メモリを使用する場合は、ASIC 標準のテスト手法を使用するため、テスト配線が必要になります。テスト手法については、**CB-9 ファミリ VX/VM タイプ 設計マニュアル (A12745J)** を参照してください。

図3-9 RAMアクセス・タイミング

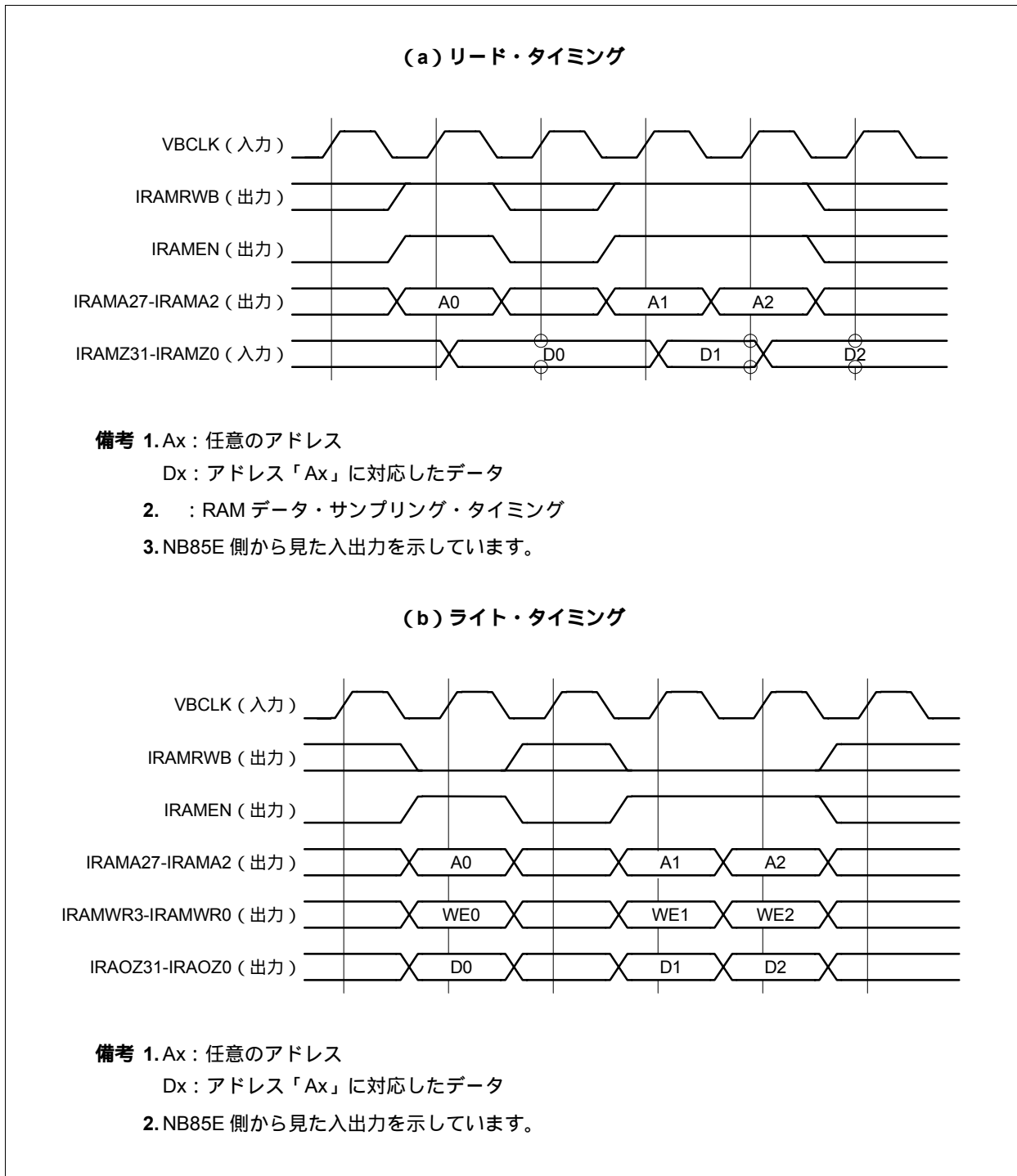
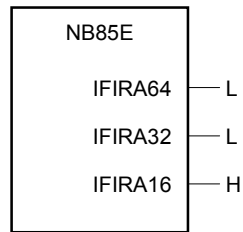


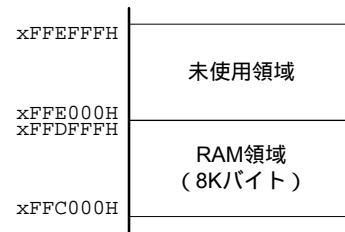
図3 - 10 NB85EのRAM領域設定

(1) 8ビット×2KワードのコンパイルドRAMを4個接続した場合

(a) 端子設定



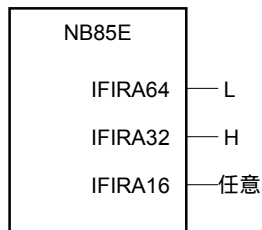
(b) メモリ・マップ



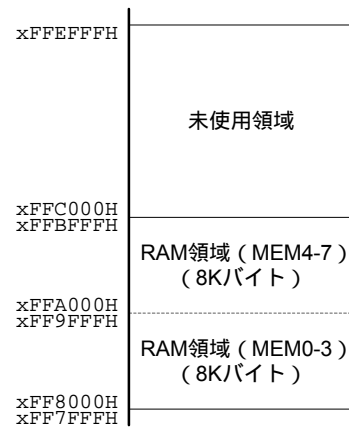
備考 L : ロウ・レベルを入力
H : ハイ・レベルを入力

(2) 8ビット×2KワードのコンパイルドRAMを8個接続した場合

(a) 端子設定

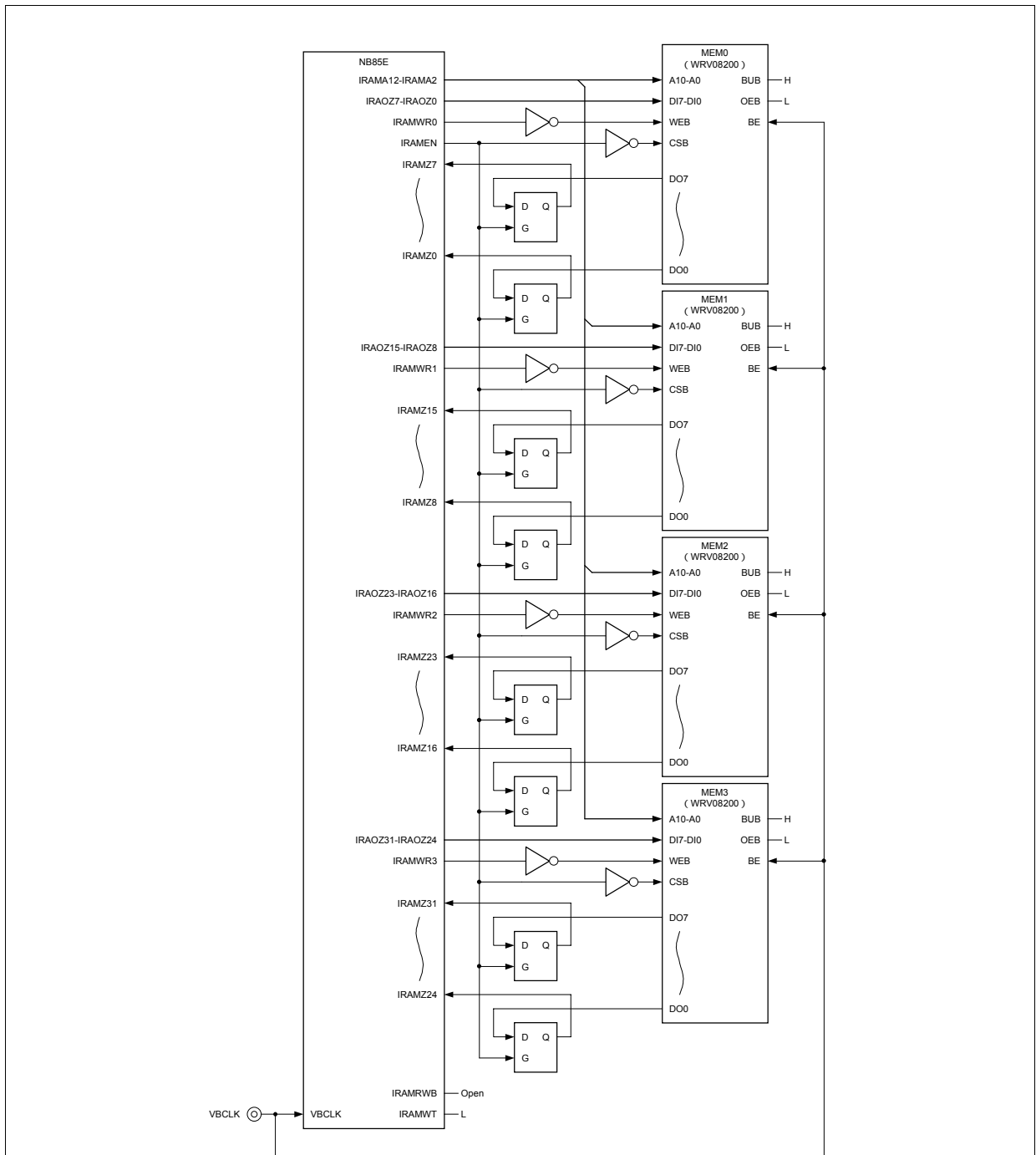


(b) メモリ・マップ



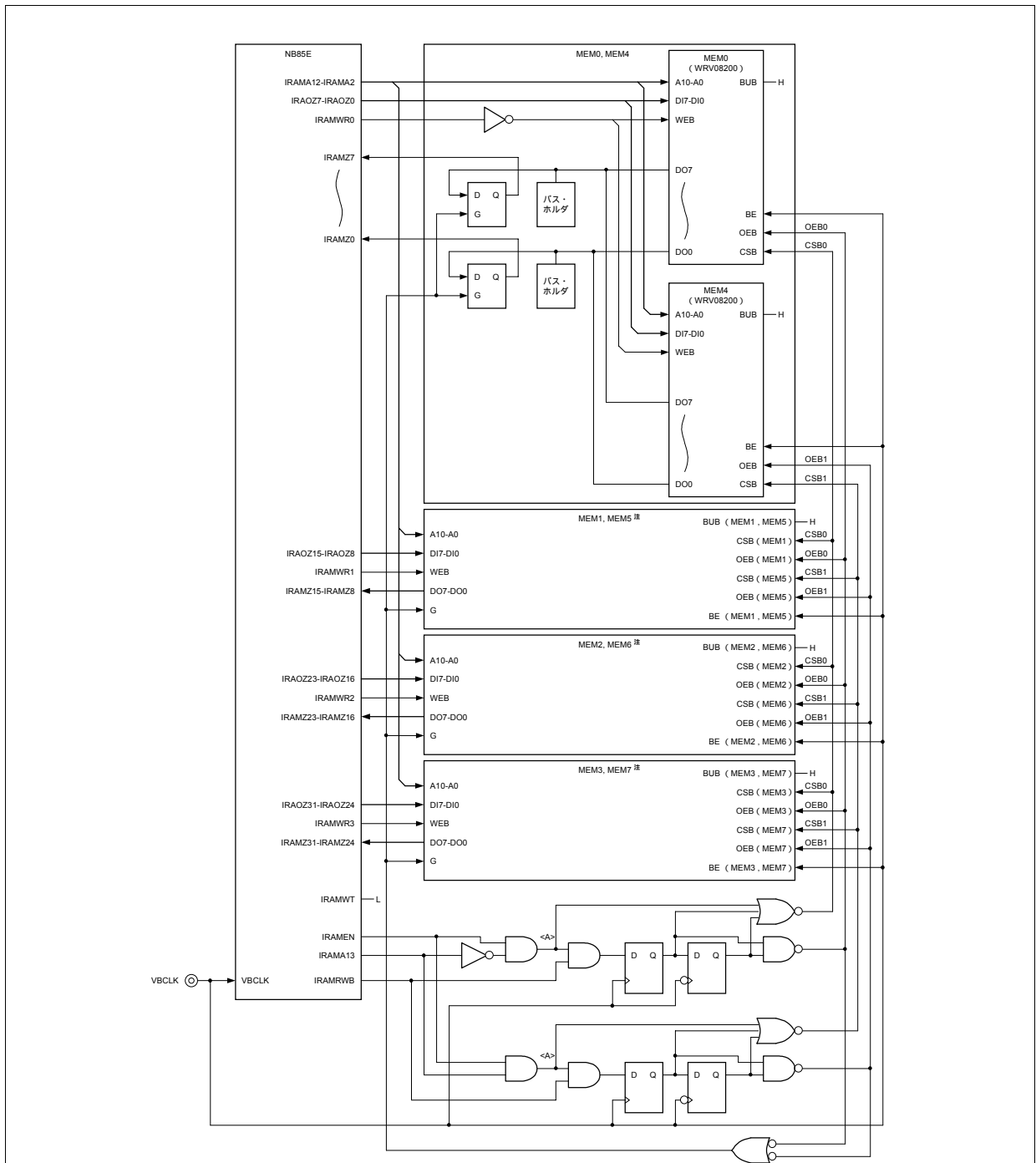
備考 L : ロウ・レベルを入力
H : ハイ・レベルを入力

図3 - 11 VDBへのコンパイルド・メモリの接続例（4個接続した場合）



- 備考 1.** この接続例では、バックアップ・モードを使用しない設定になっています。バックアップ・モードを使用する場合は、電源分離などの対応が必要となります。詳細については、**CB-9 ファミリ VX/VM タイプ 設計マニュアル メモリ・マクロ (コンパイルド・タイプ) 編 (A12982J)** を参照してください。
- 2.** データ・バスは実際のデータに対して論理が反転しているため、シミュレーションでデータをダンプする場合は注意してください。
- 3.** 図はノーマル端子の配線方法を示しています。テスト用端子は、ASIC のテスト手法に準拠して配線してください（詳細については、**CB-9 ファミリ VX/VM タイプ 設計マニュアル (A12745J)** を参照してください）。
- 4.** L : ロウ・レベルを入力 H : ハイ・レベルを入力 Open : オープン

図3 - 12 VDBへのコンパイルド・メモリの接続例（8個接続した場合）



注 MEM0, MEM4 と同じ回路構成です。

備考 1. この接続例では、バックアップ・モードを使用しない設定になっています。バックアップ・モードを使用する場合は、電源分離などの対応が必要となります。詳細については、**CB-9 ファミリー VX/VM タイプ 設計マニュアル メモリ・マクロ (コンパイルド・タイプ) 編 (A12982J)** を参照してください。

2. データ・バスは実際のデータに対して論理が反転しているので、シミュレーションでデータをダンプする場合は注意してください。

3. 図はノーマル端子の配線方法を示しています。テスト用端子は、ASIC のテスト手法に準拠して配線してください（詳細については、**CB-9 ファミリー VX/VM タイプ 設計マニュアル (A12745J)** を参照してください）。

4. L: ロウ・レベルを入力 H: ハイ・レベルを入力

3.3.1 RAM を 8 個接続した場合の動作

(1) 同一 RAM 領域からのリード

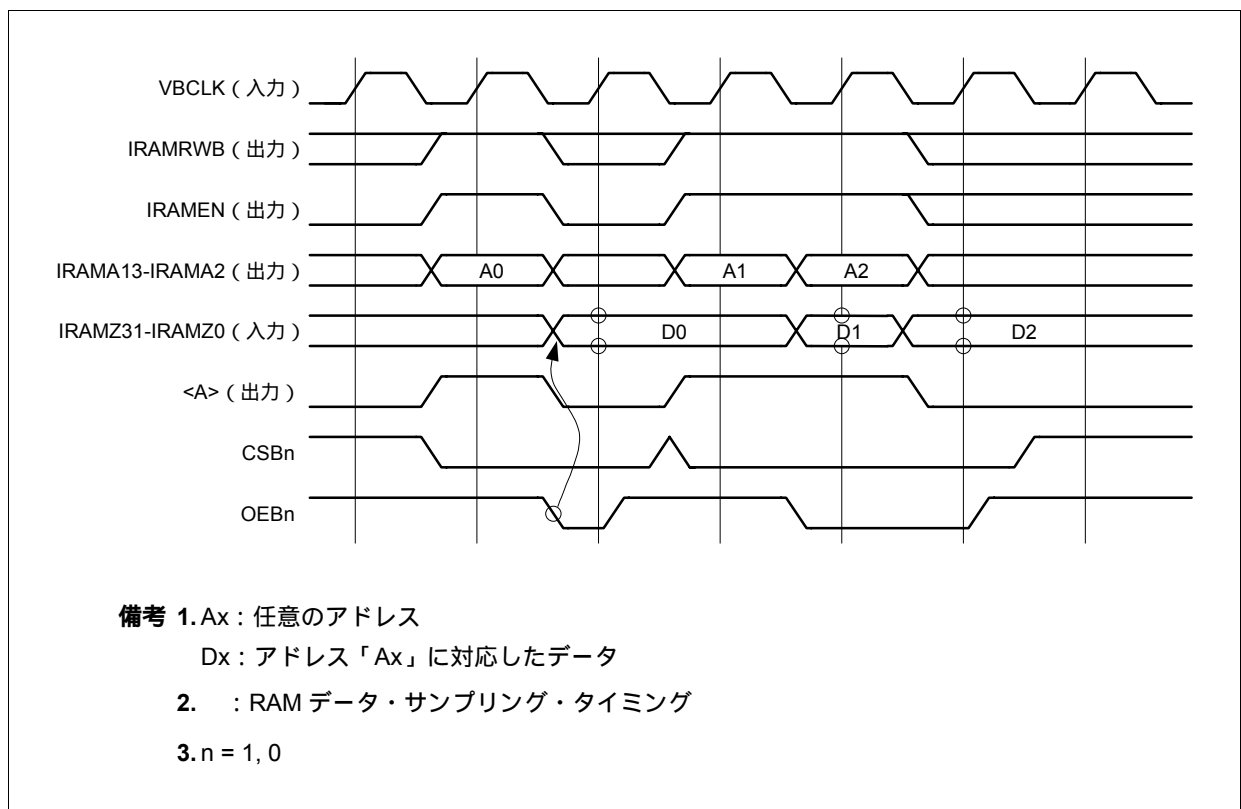
図 3 - 13 に MEM0-MEM3, または MEM4-MEM7 のどちらか一方の RAM 領域をリードしたときのタイミングを示します。VDB のパイプライン形式のリード・アクセスに対応するため, IRAMA13 をデコードした結果を 1 クロック遅らせて OEB を生成しています。CSB はデータの衝突を防ぐために, アドレスをデコードした結果と OEB の論理和で生成しています。

IRAMEN と IRAMA13 を入力し, デコードした結果 (図 3 - 12 の<A>参照) がアクティブになるとすぐに CSBn がアクティブになります。次のクロックの立ち上がりで RAM を読み出しますが, このときには OEBn がインアクティブのため出力はされません。次のクロックの立ち下がりで OEBn がアクティブになり, IRAMZ31-IRAMZ0 に有効なデータが入力されます ($n = 1, 0$)。

デコードした結果がインアクティブになった次の立ち上がりで, OEBn がインアクティブになると同時に IRAMZ31-IRAMZ0 はそのときの値を保持します。これは OEBn がインアクティブの間, RAM に対してリードを行わずに値を保持するためです。OEBn がインアクティブになった次のクロックの立ち下がり, CSBn はインアクティブになります。

なお, ライト・アクセスの場合はパイプライン形式ではないため, IRAMA13 をデコードした結果をそのまま CSB にしています。

図3 - 13 RAMリード・タイミング (同一RAM領域からのリード)



(2) RAM 領域が切り替わった場合のリード

図3-14にA0, A1がMEM0-MEM3領域, A2がMEM4-MEM7領域で, その間を連続リード・アクセスしたときのCSB0, OEB0, CSB1, OEB1のタイミングを示します。

IRAMA13-IRAMA2がA1からA2になり, RAMのアドレスがMEM0-MEM3領域からMEM4-MEM7領域に切り替わってもA1に対するデータ(D1)を読み込むタイミングは0.5クロック後のクロックの立ち上がりであるため, MEM0-MEM3領域用のOEB0を0.5クロック, CSB0を1クロック伸ばしています。CSB1はD2を読み込むために, IRAMA13-IRAMA2がA2になったときにアクティブになります。OEB1はデータの衝突を防ぐため, OEB0がインアクティブになってから0.5クロック後にアクティブにしています。

図3-14 RAMリード・タイミング (RAM領域が切り替わった場合)

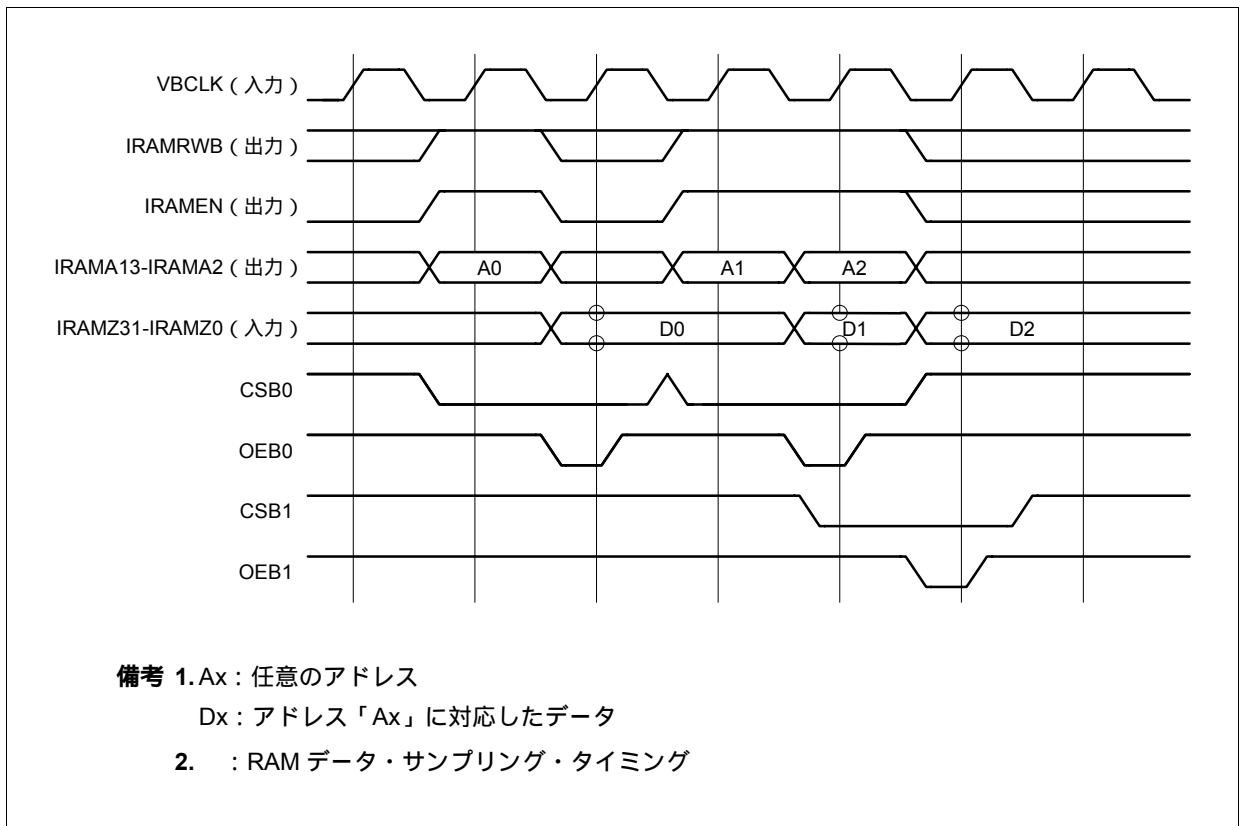
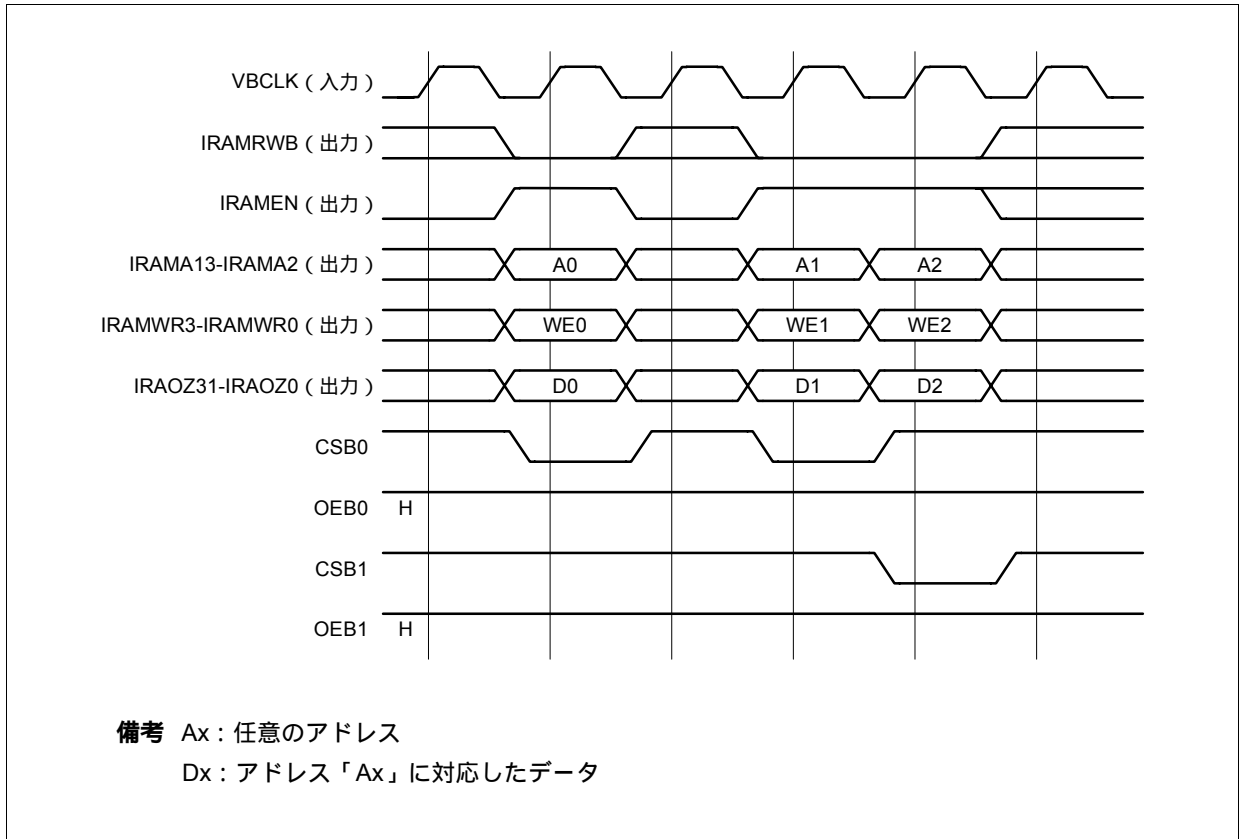


図3 - 15 に A0, A1 が MEM0-MEM3 領域, A2 が MEM4-MEM7 領域で, その間を連続ライト・アクセスしたときの CSB0, CSB1 のタイミングを示します。

ライト・アクセスの場合はパイプライン処理を行う必要がないため, IRAMEN とアドレスをデコードした結果で CSB を生成します。

図3 - 15 RAMライト・タイミング



第 4 章 VSB への接続

4.1 ユーザ・ロジックの接続

この節では、VSB にユーザ・ロジックを接続する方法について説明します。

ユーザ・ロジックとして、3 種類の読み書き可能な 32 ビットのレジスタを例とします。これらのレジスタは、それぞれ、ノー・ウエイト/1 ウエイト/アドレス・ホールドで動作します。

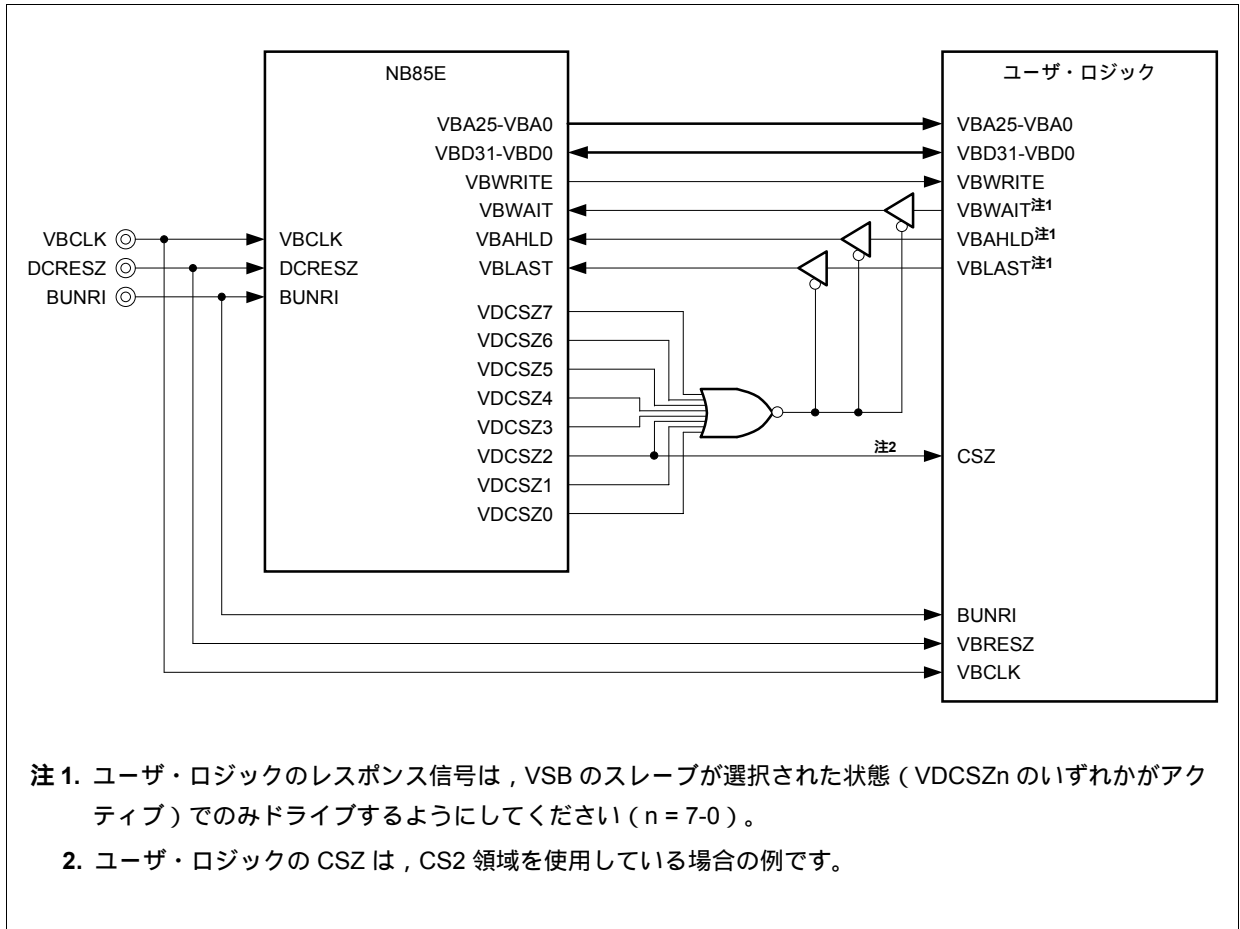
なお、各レジスタは、ワード単位でだけ書き込みが可能で、バイトやハーフワード命令で書き込むと対象外のビットには不定値が書き込まれます。また、リセット入力ですべてのビットがクリア (0) されます。

ユーザ・ロジックが外部とのインタフェースに使用する信号は次のとおりです。

表4-1 ユーザ・ロジックのインタフェース信号

信号名	入出力	機能
VBCLK	入力	システム・クロック
VBRESZ	入力	システム・リセット
CSZ	入力	セレクト信号 (VDCSZ7-VDCSZ0 のうちの 1 本)
VBA25-VBA0	入力	アドレス入力
VBD31-VBD0	入出力	データ入出力
VBWRITE	入力	ライト・ステータス
VBWAIT	出力	ウエイト・レスポンス出力
VBAHLD	出力	アドレス・ホールド・レスポンス出力
VBLAST	出力	ラスト・レスポンス出力
BUNRI	入力	ノーマル/テスト・モードの選択入力 (テスト時に出力をハイ・インピーダンス状態にするために使用します)

図4-1 VSBへのユーザ・ロジック接続例



4.1.1 VSB 動作概要

VSB の基本タイミングは、図4-2に示すように1クロック・アクセスとなっています。また、アドレスに対するデータが1クロック・シフトする「パイプライン形式」のバスになっています。

必要に応じてVBWAIT 信号やVBAHLD 信号を操作し、ウエイト・サイクルやアドレス・ホールド・サイクルを挿入してタイミング調整を行います。

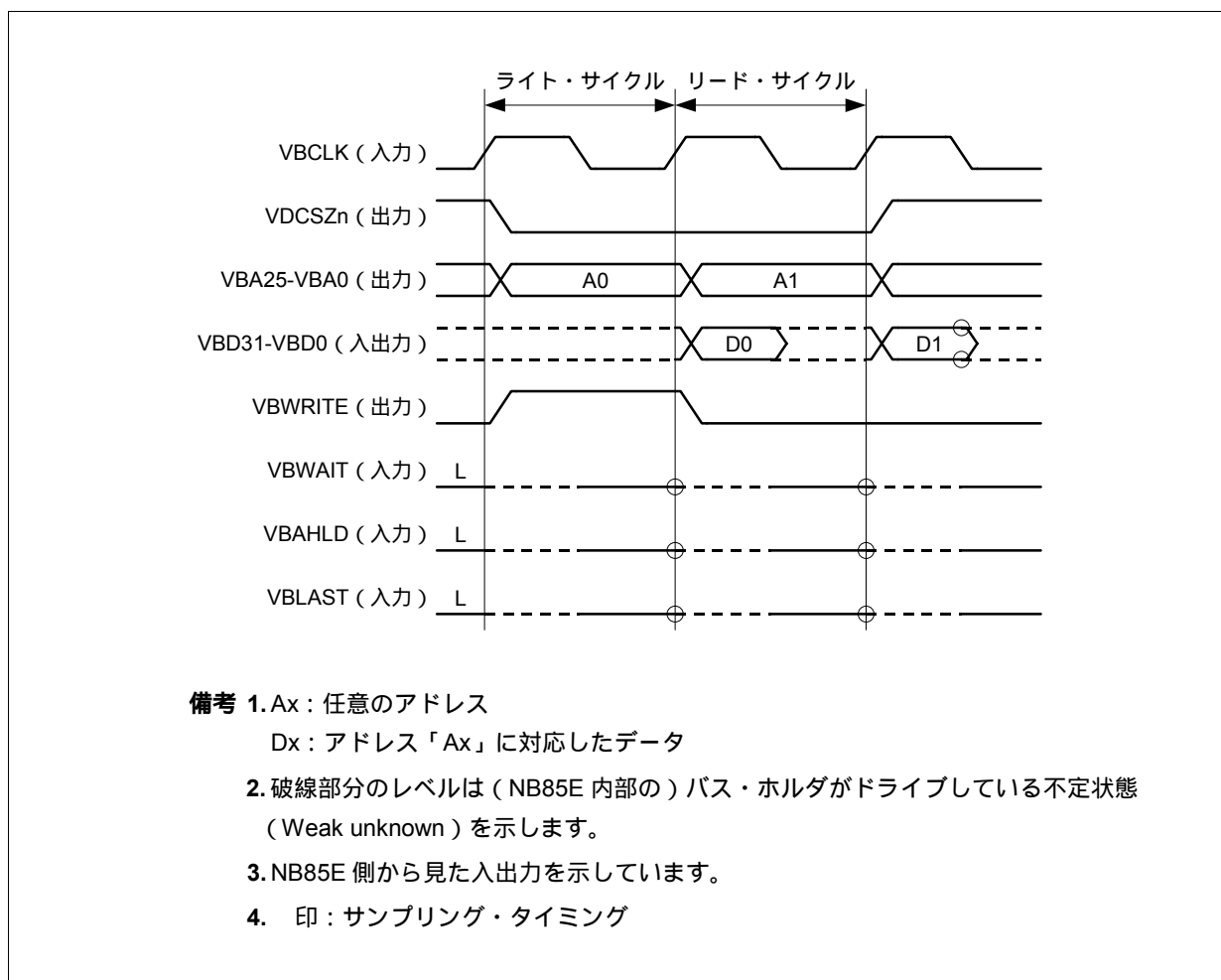
(1) ノー・ウエイト時（基本タイミング）

アドレスを1クロック間出力し、次の半クロック間でリードまたはライトを行います。

データ・バス（VBD31-VBD0）は、クロック（VBCLK）がハイ・レベルの間だけドライブします。クロックがロウ・レベルの間は、NB85E 内部のバス・ホルダがドライブしている不定状態（Weak unknown）となります。

VBWAIT, VBAHLD, VBLAST 信号は、クロックがロウ・レベルの間だけドライブします。クロックがハイ・レベルの間は、NB85E 内部のバス・ホルダがドライブしている不定状態（Weak unknown）となります。

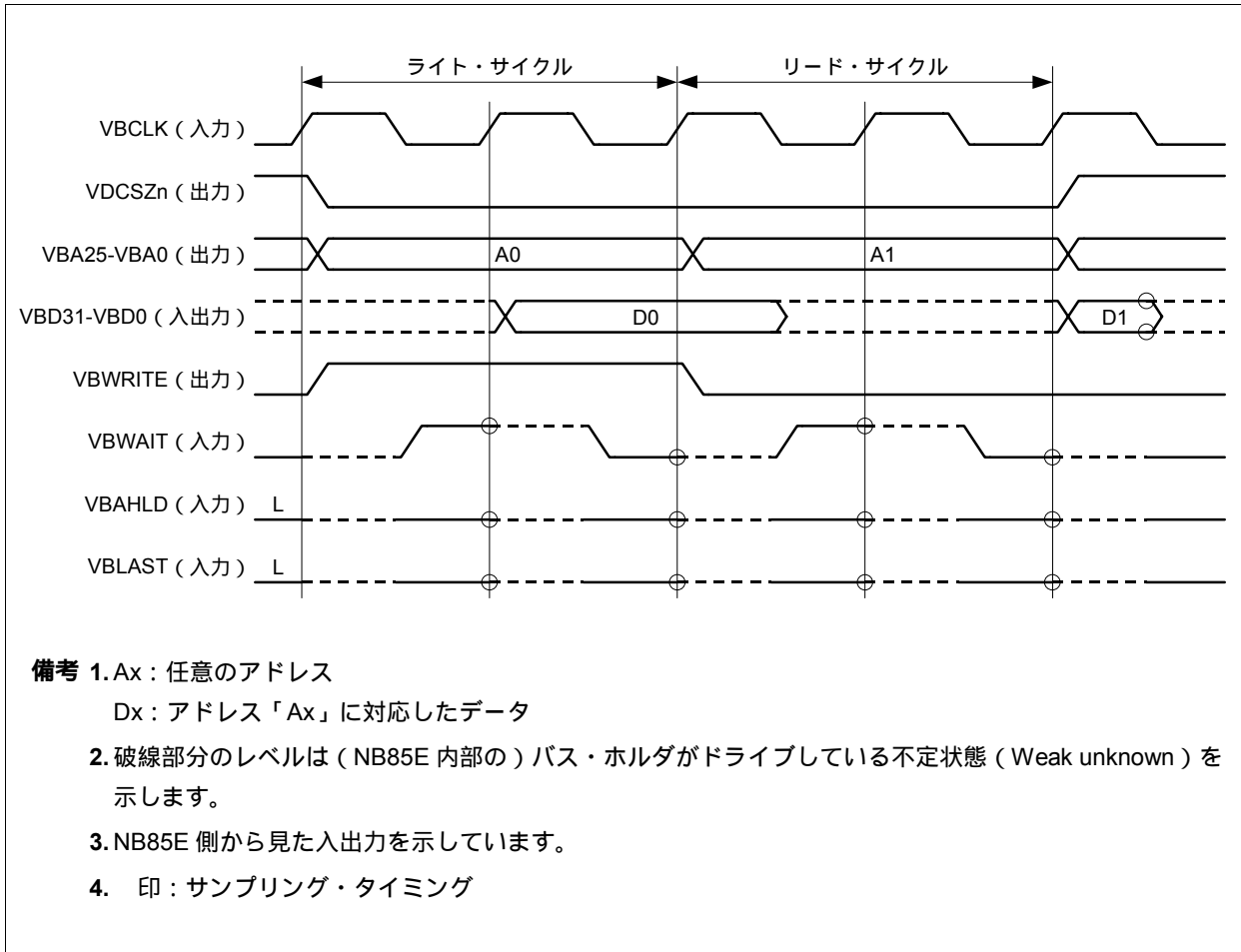
図4-2 VSBタイミング（ノー・ウエイト）



(2) 1 ウエイト時

アドレスを2クロック期間出力し、次の半クロックでリード、またはライトを行います。ウエイトを挿入することによって、低速な回路を接続できます。

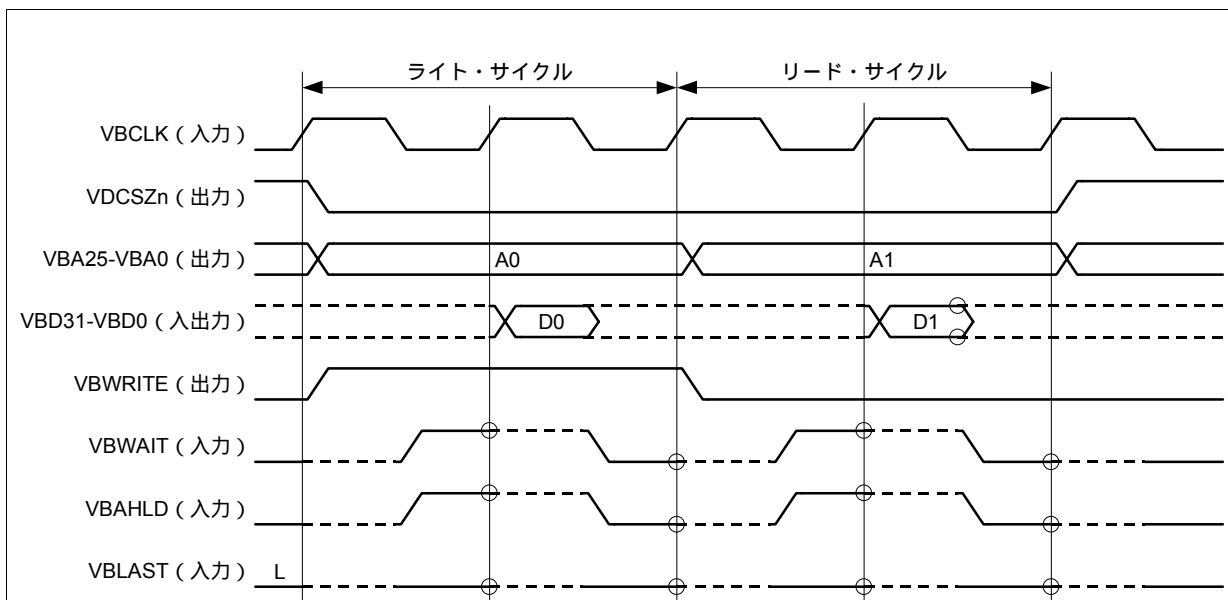
図4-3 VSBタイミング(1ウエイト)



(3) アドレス・ホールド時

1 ウェイト時と同様に、2 クロック期間で1つのデータを転送します。アドレス出力からデータ入出力までの時間は基本タイミングと同じですが、リード・サイクルやライト・サイクルの終了までアドレスが保持されるため、インタフェース回路の設計が容易になります。

図4-4 VSBタイミング (アドレス・ホールド)



備考 1. Ax : 任意のアドレス

Dx : アドレス「Ax」に対応したデータ

2. 破線部分のレベルは (NB85E 内部の) バス・ホルダがドライブしている不定状態 (Weak unknown) を示します。

3. NB85E 側から見た入出力を示しています。

4. 印 : サンプルング・タイミング

4.1.2 回路例

(1) ノー・ウエイト時

ノー・ウエイトで動作するユーザ・ロジックの接続例を示します。この例では、高速にリード/ライトが行えますが、VBCLK の周波数が高い場合、データのセットアップ時間やホールド時間を確保するのが困難です。

図4 - 5 ノー・ウエイトで動作するユーザ・ロジックのHDL作成例

```

module UDL_NOWAIT( VBCLK, VBRESZ, CSZ, VBA, VBD, VBWRITE, VBWAIT, VBAHLD, VBLAST, BUNRI );
  input      VBCLK ;
  input      VBRESZ ;
  input      CSZ ;
  input [25:0] VBA ;
  inout [31:0] VBD ;
  input      VBWRITE ;
  output     VBWAIT ;
  output     VBAHLD ;
  output     VBLAST ;
  input      BUNRI ;

  // nowait
  reg [31:0] reg1 ;
  reg        sel_reg_rd ;
  reg        sel_reg_wr ;

  wire       sel = (CSZ==1'b0 && VBA[7:2]==6'b0000_00) ? 1'b1 : 1'b0 ;

  // wait control
  assign     VBWAIT = (~VBCLK & sel) ? 1'b0 : 1'bZ ;
  assign     VBAHLD = (~VBCLK & sel) ? 1'b0 : 1'bZ ;
  assign     VBLAST = (~VBCLK & sel) ? 1'b0 : 1'bZ ;

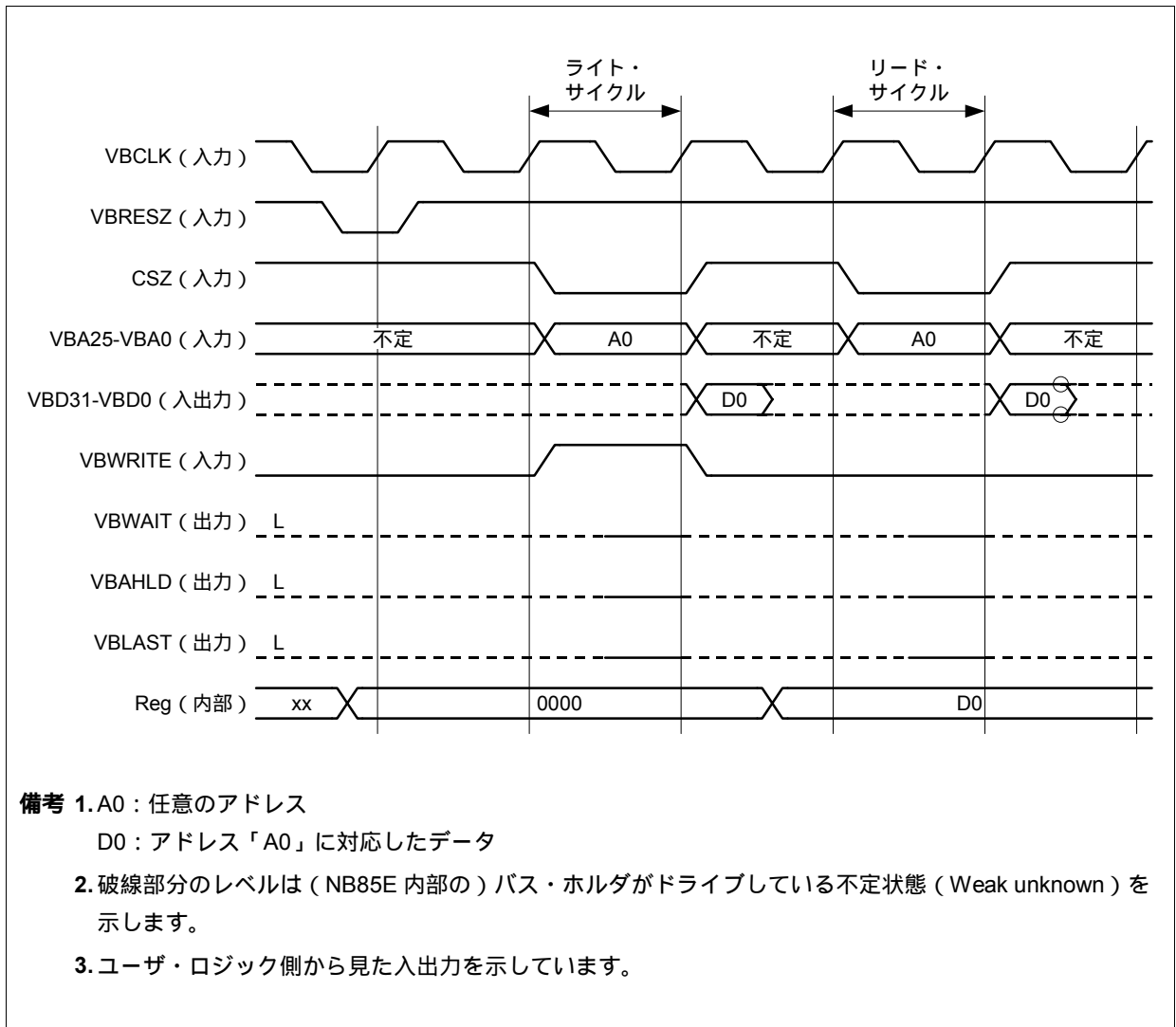
  // read
  assign     VBD = ( sel_reg_rd & VBCLK & ~BUNRI ) ? reg1 : 32'hZZZZZZZZ ;

  always @( VBCLK )
  begin
    if( VBCLK == 1'b0 )
    begin
      sel_reg_rd <= sel & ~VBWRITE ;
    end
  end

  // write
  always @( negedge VBCLK or negedge VBRESZ )
  begin
    if( ~VBRESZ )
      reg1 <= 32'h00000000 ;
    else
    begin
      if( sel_reg_wr )
        reg1 <= VBD ;
      sel_reg_wr <= sel & VBWRITE ;
    end
  end
end
endmodule

```

図4-6 ノー・ウエイト時のタイミング・チャート



(2) 1 ウェイト挿入時

1 ウェイトで動作するユーザ・ロジックの接続例を示します。この例では、データのリード/ライト・タイミングを1クロック遅らせることにより、アクセス・タイムに余裕を持つことができます。

なお、ウェイト挿入数を増やす場合、VBSTZ 信号をシフト・レジスタのような構成で伸ばして使用してください。

図4-7 1ウェイトで動作するユーザ・ロジックのHDL作成例

```

module UDL_1WAIT( VBCLK, VBRESZ, CSZ, VBA, VBD, VBSTZ, VBWRITE, VBWAIT, VBAHLD, VBLAST, BUNRI );
  input          VBCLK;
  input          VBRESZ;
  input          CSZ;
  input [25:0]   VBA;
  inout [31:0]  VBD;
  input          VBSTZ;
  input          VBWRITE;
  output         VBWAIT;
  output         VBAHLD;
  output         VBLAST;
  input          BUNRI;

  reg [31:0]     reg2;
  reg            sel_reg_rd;
  reg            sel_reg_wr;

  wire          sel = (CSZ==1'b0 && VBA[7:2]==6'b0000_01) ? 1'b1 : 1'b0;

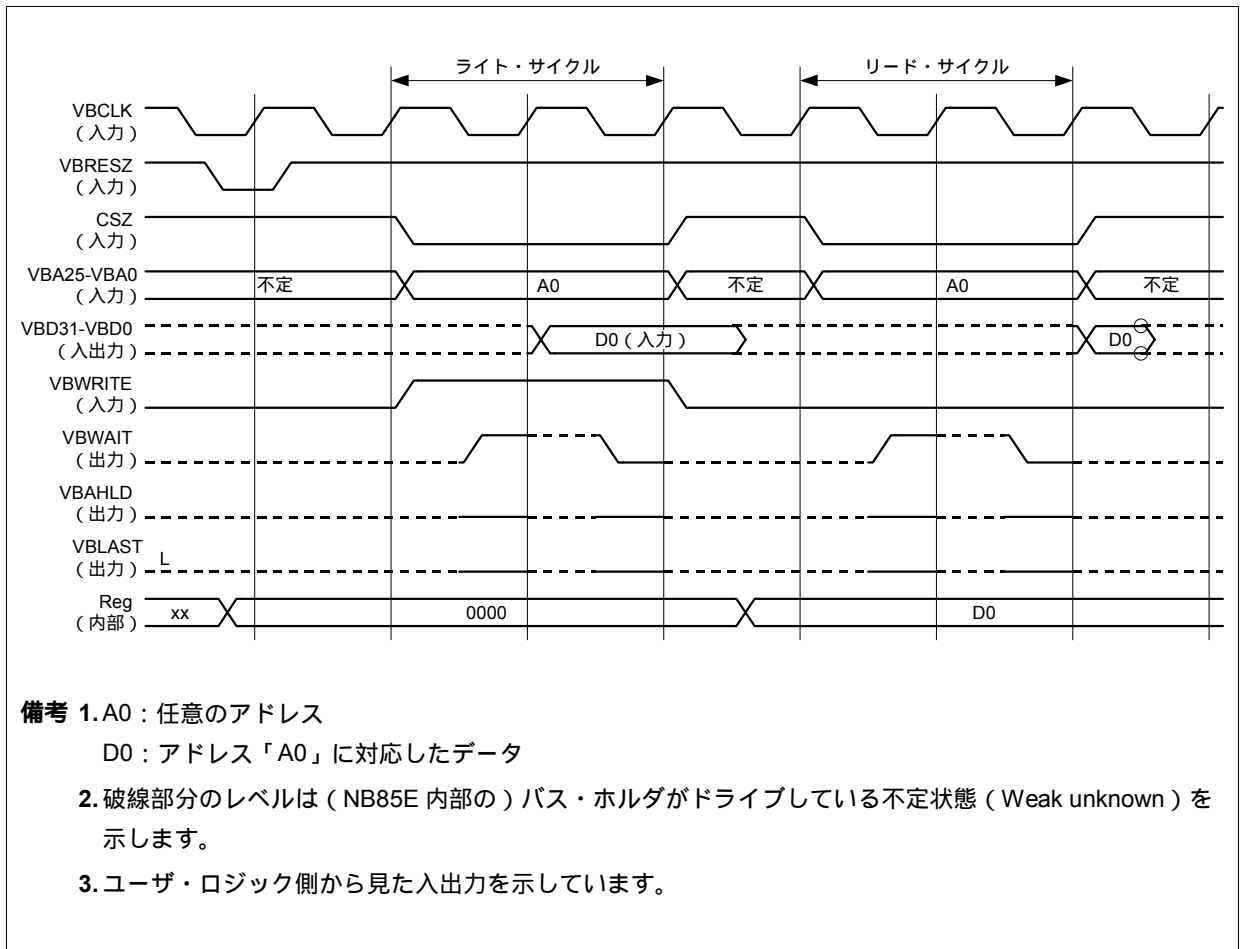
// wait control
  wire          wa = ~VBSTZ; // 1wait
  assign        VBWAIT = (~VBCLK & sel) ? wa : 1'bZ;
  assign        VBAHLD = (~VBCLK & sel) ? 1'b0 : 1'bZ;
  assign        VBLAST = (~VBCLK & sel) ? 1'b0 : 1'bZ;

// read
  assign        VBD = ( sel_reg_rd & VBCLK & ~BUNRI ) ? reg2 : 32'hZZZZZZZZ;
  always @( VBCLK )
  begin
    if( VBCLK == 1'b0 )
    begin
      sel_reg_rd <= sel & ~VBWRITE;
    end
  end

// write
  always @( negedge VBCLK or negedge VBRESZ )
  begin
    if( ~VBRESZ )
      reg2 <= 32'h00000000;
    else
    begin
      if( sel_reg_wr )
        reg2 <= VBD;
      sel_reg_wr <= sel & VBWRITE & ~wa;
    end
  end
end
endmodule

```

図4-8 1ウェイト挿入時のタイミング・チャート



(3) アドレス・ホールド時

アドレス・ホールド機能を使用したユーザ・ロジックの接続例を示します。この例では、データのリード/ライト・サイクルでも、アドレスが変化しません。そのため、アドレスをラッチする必要がなく、回路を簡素化できます。

また、ウエイト挿入と組み合わせて、アクセス・タイムの長いI/Oの接続も可能です。

図4-9 アドレス・ホールド機能を使用したユーザ・ロジックのHDL作成例

```

module UDL_AHLD( VBCLK, VBRESZ, CSZ, VBA, VBD, VBSTZ, VBWRITE, VBWAIT, VBAHLD, VBLAST, BUNRI );
  input          VBCLK;
  input          VBRESZ;
  input          CSZ;
  input [25:0]   VBA;
  inout [31:0]  VBD;
  input          VBSTZ;
  input          VBWRITE;
  output         VBWAIT;
  output         VBAHLD;
  output         VBLAST;
  input          BUNRI;

  reg [31:0]     reg3;

  wire           sel = (CSZ==1'b0 && VBA[7:2]==6'b0000_10) ? 1'b1 : 1'b0;

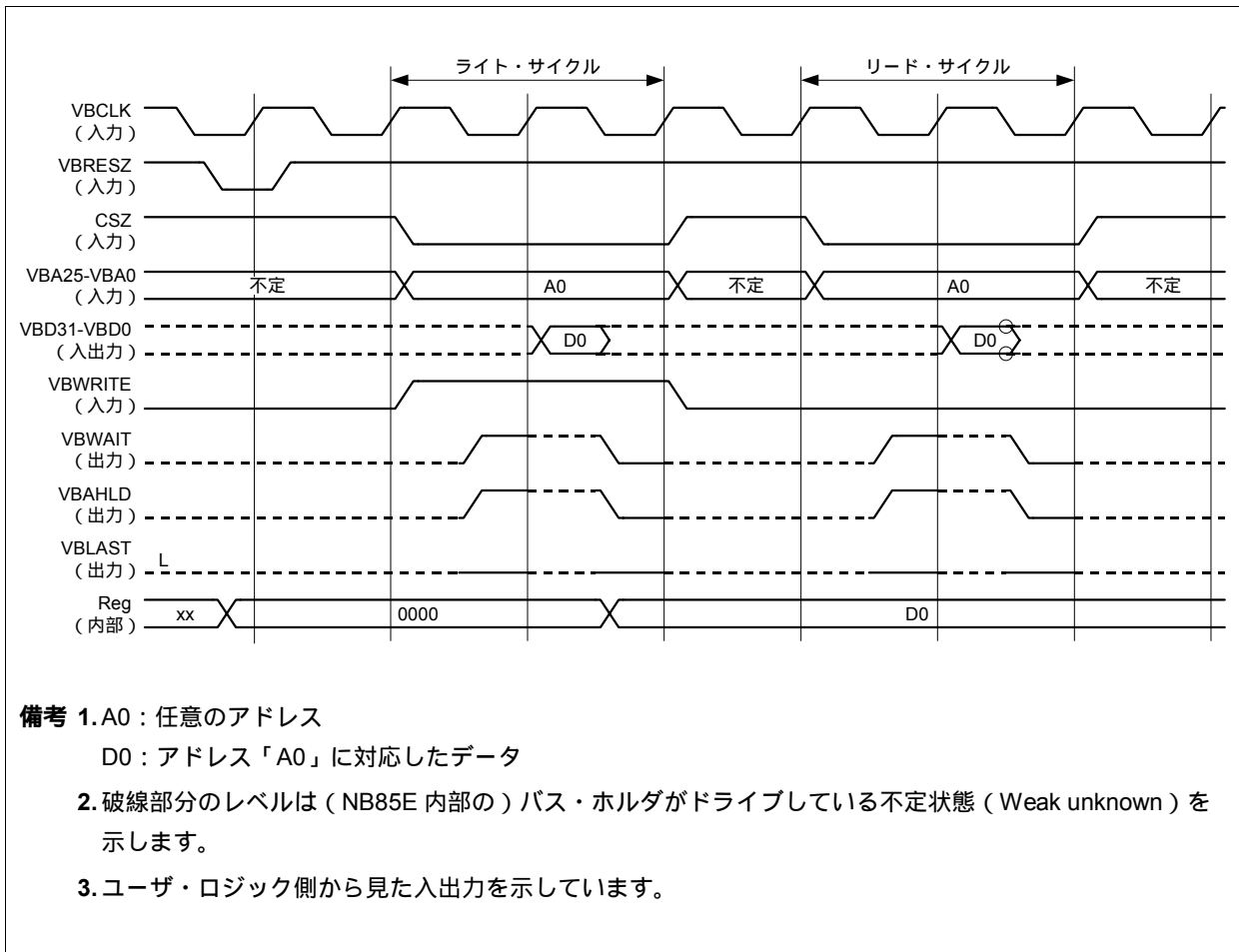
// wait control
  wire           wa = ~VBSTZ; // 1wait
  assign         VBWAIT = (~VBCLK & sel) ? wa : 1'bZ;
  assign         VBAHLD = (~VBCLK & sel) ? wa : 1'bZ;
  assign         VBLAST = (~VBCLK & sel) ? 1'b0 : 1'bZ;

// read
  reg            VBSTZ_1E
  always @(posedge VBCLK)
    VBSTZ_1E = VBSTZ;
  wire           sel_reg_rd = (sel & ~VBWRITE);
  assign         VBD = (sel_reg_rd & VBCLK & ~VBSTZ_1E & BUNRI) ? reg3 : 32'hZZZZZZZZ;

// write
  wire           sel_reg_wr = (sel & VBWRITE & ~wa);
  always @(negedge VBCLK or negedge VBRESZ)
  begin
    if( ~VBRESZ )
      reg3 <= 32'h00000000;
    else
      begin
        if( sel_reg_wr )
          reg3 <= VBD;
      end
    end
  end
endmodule

```

図4 - 10 アドレス・ホールド時のタイミング・チャート



4.2 コンパイルド・メモリの接続

この節では、メモリ・コントローラ (MEMC) を使用せずに、高密度同期式 1 ポート RAM を VSB に接続する方法を説明します。

図 4 - 11 に接続例、図 4 - 12 にタイミング・チャート、図 4 - 13 にコンパイルド・メモリ・アクセス制御回路 (VSB_VDL_MEMCBC9) の HDL 作成例を示します。

この接続例では、次に示す高密度同期式 1 ポート RAM を使用しています。

- マクロ・ブロック名 : W8K08100
- メモリ容量 (合計) : 4K バイト (1024 ワード × 8 ビット × 4[※])
- 動作周波数 : 66MHz

注 高密度同期式 1 ポート RAM のビット幅は、1 ビット単位で 1-32 ビットから選択できます。VSB のデータ・バス幅は 32 ビットですが、RAM の場合は、8 ビット単位でのアクセスを可能にする必要があります。そのため、ビット幅が 8 ビットの RAM を 4 つ接続し、32 ビット幅を構成します。

なお、VSB は通常アクセス時に、ある任意のアドレスに対するデータが 1 サイクルだけシフトする「パイプライン方式」のバス動作を行います。図 4 - 11 の接続例では、VBWAIT, VBAHLD, VBLAST 信号を制御し、リード時にはアドレス・ホールド状態で、ライト時にはウエイト状態で NB85E に対して応答することで、コンパイルド・メモリのアクセス制御を簡単化しています。

また、CB-9 ファミリ VX/VM タイプの同期式 RAM には、次のタイミング制限があります。

- クロック入力信号 (BE) の立ち上がりと同タイミングでのアドレス変化の禁止
- クロック入力信号 (BE) がハイ・レベルの状態での CSB 信号変化の禁止

これらの制限を守るため、図 4 - 11 の接続例では次のような処置を行っています。

- クロック入力端子 (BE) には、NB85E へのクロック (VBCLK) を入力
- VBCLK の立ち上がりで変化するアドレスと CSB (VDCSZ) を VBCLK の立ち下がりでラッチして使用
- OEB も CSB と同じタイミングになるようにラッチして使用
- WEB は、CSB = 0, OEB = 1 の条件での NB85E からのバイト・イネーブル信号である VBBENZ 信号を使用

図4 - 11 コンパイルド・メモリの接続例

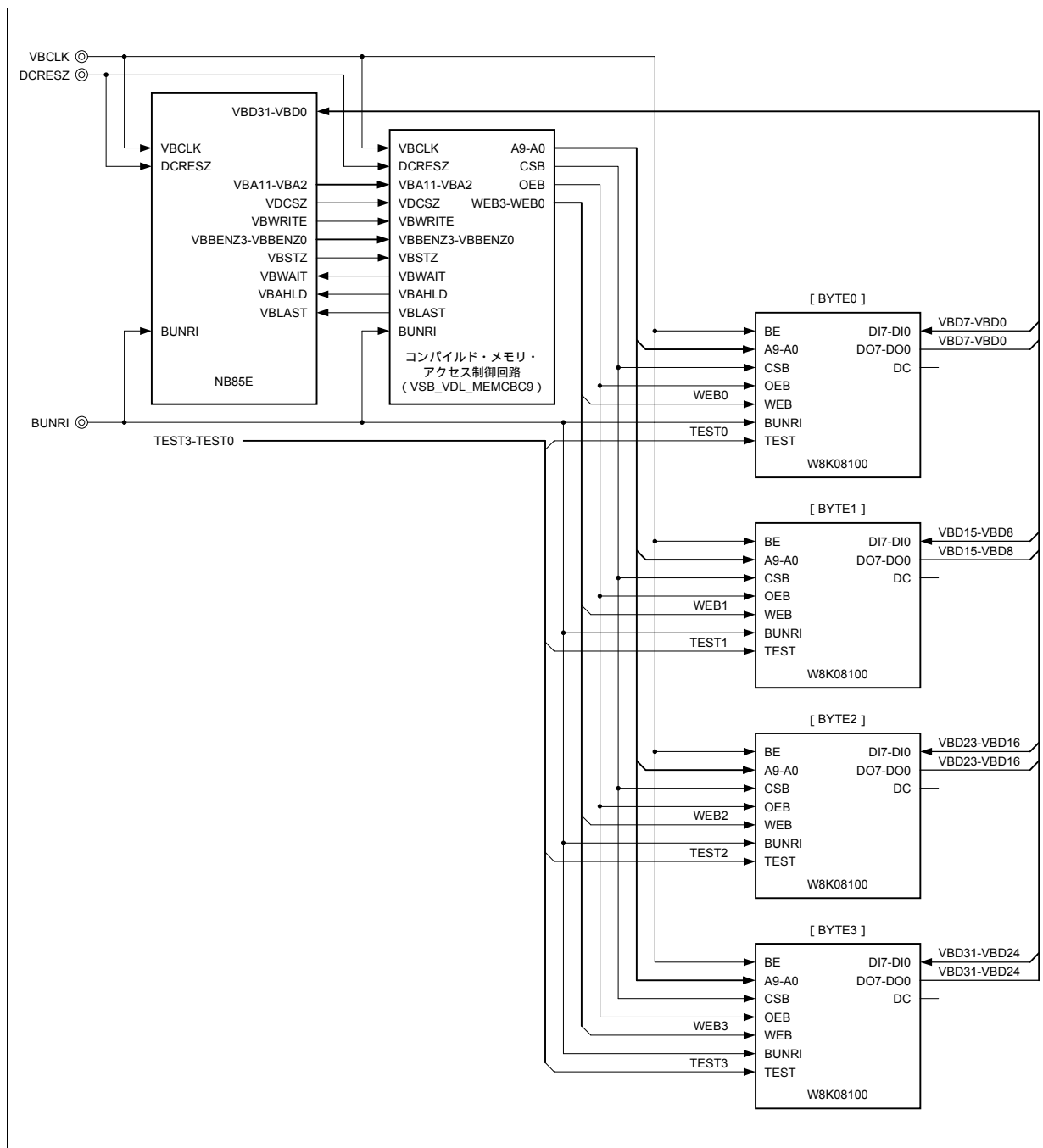


図4 - 12 コンパイルド・メモリ・アクセス時のタイミング・チャート

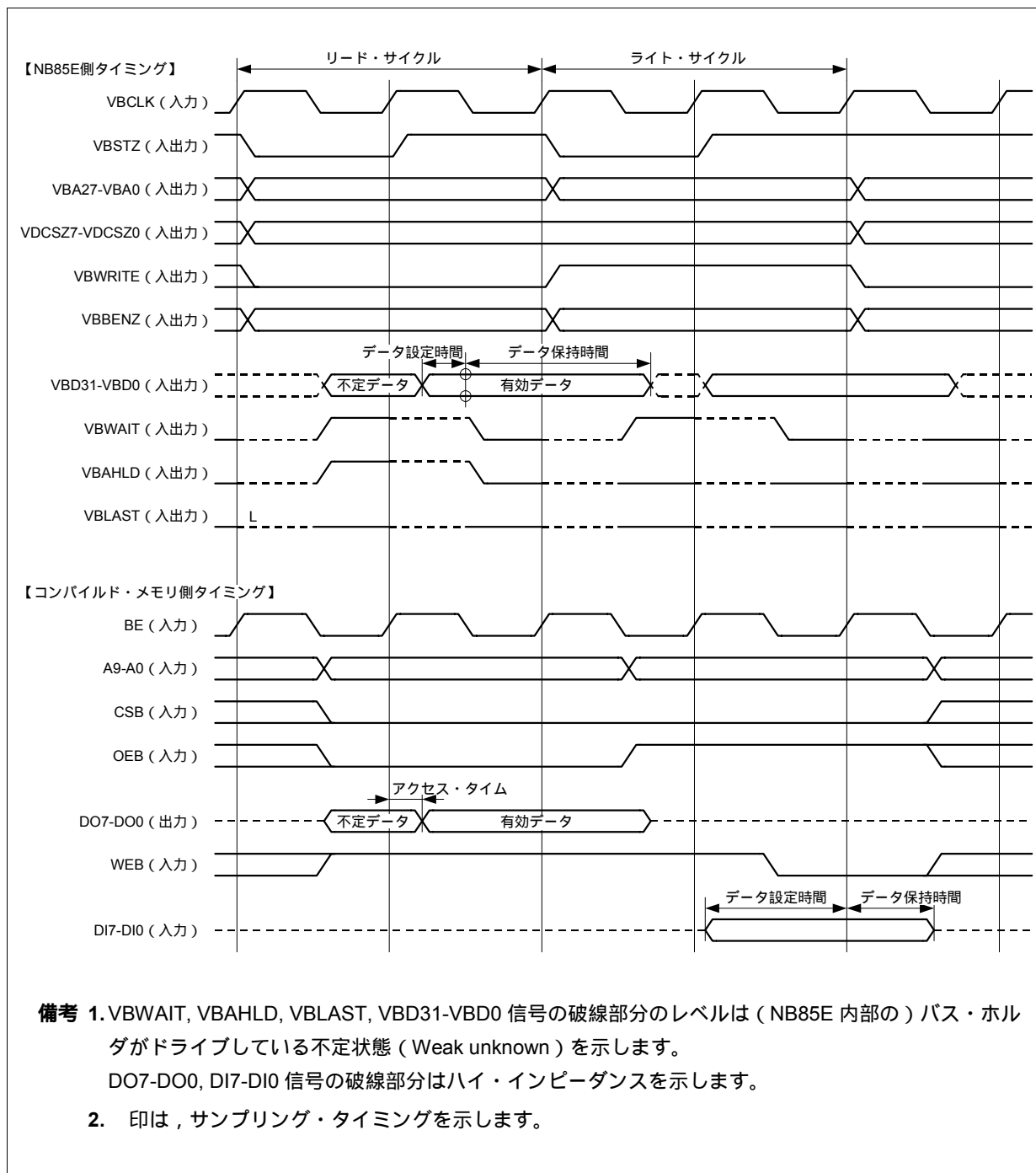


図4 - 13 コンパイルド・メモリ・アクセス制御回路のHDL作成例 (1/2)

```

module VSB_UDL_MEMCBC9 (
  VPRESZ,VBCLK,VBA,A,VDCSZ,VBWRITE,CSB,OEB,VBBENZ,WEB,VBSTZ,
  VBWAIT,VBAHLD,VBLAST,BUNRI );

  input      VPRESZ ;      // Reset
  input      VBCLK ;      // System clock
  input [11:2] VBA ;      // Address bus from VSB
  output [9:0] A ;        // Address bus to Memory macro
  input      VDCSZ ;      // Chip area select
  input      VBWRITE ;    // Transfer direction
  input [3:0] VBBENZ ;    // Byte enable
  output     CSB ;        // Chip area select
  output     OEB ;        // RAM data output enable
  output [3:0] WEB ;      // Write enable each byte
  input      VBSTZ ;      // Transfer start
  output     VBWAIT ;     // Wait response
  output     VBAHLD ;     // Address hold response
  output     VBLAST ;     // Last data response
  input      BUNRI ;      // Test mode

  wire [11:2] VBA ;
  reg  [9:0]  A ;
  reg  [3:0]  WEB ;
  reg        CSB,OEB ;
  reg        ResOut ;
  wire [3:0]  VBBENZ ;
  wire        VBWAIT,VBAHLD,VBLAST ;
  reg  [1:0]  State ;
  reg  [2:0]  Response ;
  reg        WRITE ;

  // Define status & output for Response({VBWAIT,VBAHLD,VBLAST})
  parameter [1:0] S_READY = 2'b00, S_WAIT = 2'b10, S_AHOLD = 2'b11 ;
  parameter [2:0] O_READY = 3'b000,O_WAIT = 3'b100,O_AHOLD = 3'b110 ;

  // synopsys async_set_reset "VPRESZ"
  always @( negedge VBCLK or negedge VPRESZ ) begin
    if( ~VPRESZ ) begin
      A      = 10'h0 ;
      CSB    = 1'b1 ;
      OEB    = 1'b1 ;
    end else begin
      A      = VBA ;
      CSB    = VDCSZ ;
      OEB    = VBWRITE ;
    end
  end

  always @( negedge VBCLK or negedge VPRESZ ) begin
    if( ~VPRESZ ) begin
      WEB    = 4'b1111 ;
      WRITE  = 0 ;
    end else if( ~VBSTZ && VBWRITE ) begin // write cycle 1
      WEB    = 4'b1111 ;
      WRITE  = 1 ;
    end else if( WRITE ) begin // write cycle 2
      if( ~CSB ) WEB = VBBENZ ;
      else    WEB = 4'b1111 ;
      WRITE  = 0 ;
    end else begin
      WEB    = 4'b1111 ;
      WRITE  = 0 ;
    end
  end
end

```


図4 - 13 コンパイルド・メモリ・アクセス制御回路のHDL作成例 (2/2)

```
always @( negedge VBCLK or negedge VPRESZ ) begin
    if( ~VPRESZ ) begin
        State      = S_READY ;
        Response = O_READY ;
    end else if( ~VBSTZ && ! VBWRITE ) begin          // read cycle
        State      = S_AHOLD ;
        Response = O_AHOLD ;
    end else if( ~VBSTZ && VBWRITE ) begin           // write cycle
        State      = S_WAIT ;
        Response = O_WAIT ;
    end else begin
        State      = S_READY ;
        Response = O_READY ;
    end
end

always @( VDCSZ or VBCLK or BUNRI) ResOut = ~VDCSZ & ~VBCLK & ~BUNRI ;
assign { VBWAIT,VBAHLD,VBLAST } = ResOut ? Response : 3'bzzz ;

endmodule
```

第5章 メモリ・コントローラ (MEMC) の接続

NB85E500 は NB85E, NB85ET 用のメモリ・コントローラで, NU85E500 は NB85E 用のメモリ・コントローラです。

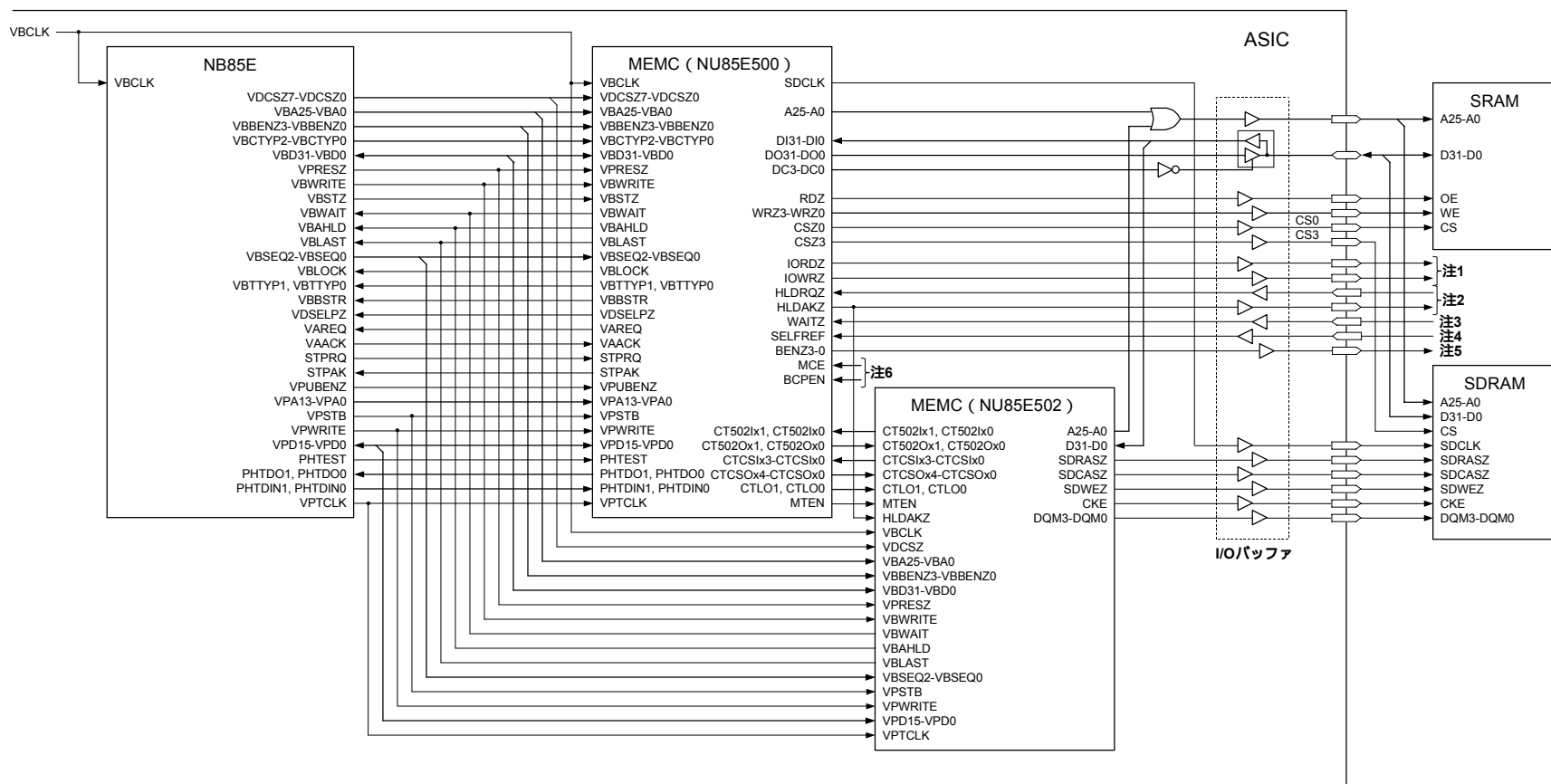
Nx85E500 は, 接続する外部メモリの種類に応じて次のように使用します。

対象 CPU コア	接続する外部メモリの種類	メモリ・コントローラ (MEMC)
NB85E	SRAM, ROM, ページ ROM, フラッシュ・メモリ	NB85E500/NU85E500
	SDRAM	NB85E500/NU85E500 + NU85E502
NB85ET	SRAM, ROM, ページ ROM, フラッシュ・メモリ	NB85E500
	SDRAM	NB85E500 + NU85E502

接続するメモリは 8/16/32 ビット幅のいずれの製品でも接続可能ですが, 16 ビット・バス幅や 32 ビット・バス幅のメモリでは, バイト単位でライトする必要があります。SRAM を接続する場合は WE 信号で, SDRAM を接続する場合は DQM 信号でバイト単位でライトします。

詳細については, **メモリ・コントローラ ユーザーズ・マニュアル NB85E, NB85ET 編 (A14206J)** を参照してください。

図5-1 NB85E, MEMC, 外部メモリ (SRAM, SDRAM) の接続例



注1. DMA フライバイ転送時に使用

2. バス・ホールド時に使用

3. 外部ウェイト制御時に使用

注4. セルフ・リフレッシュ要求時に使用

5. バイト・イネーブル使用時

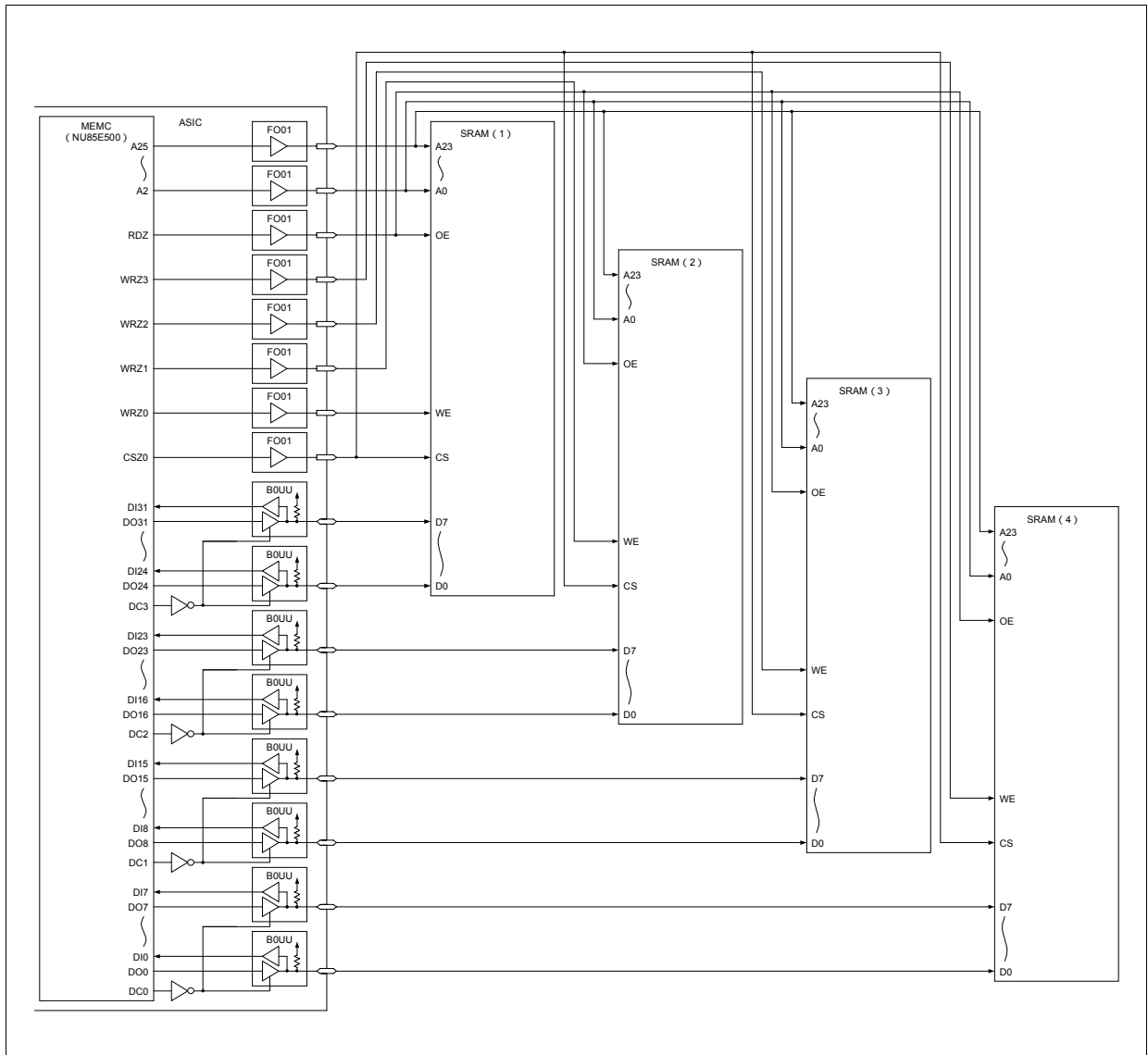
6. 初期設定用

5.1 SRAM との接続

SRAM との接続例を図 5 - 2 に示します。

この接続例では、8 ビット幅の SRAM を 4 個接続して、32 ビット幅でのアクセスを可能としています。

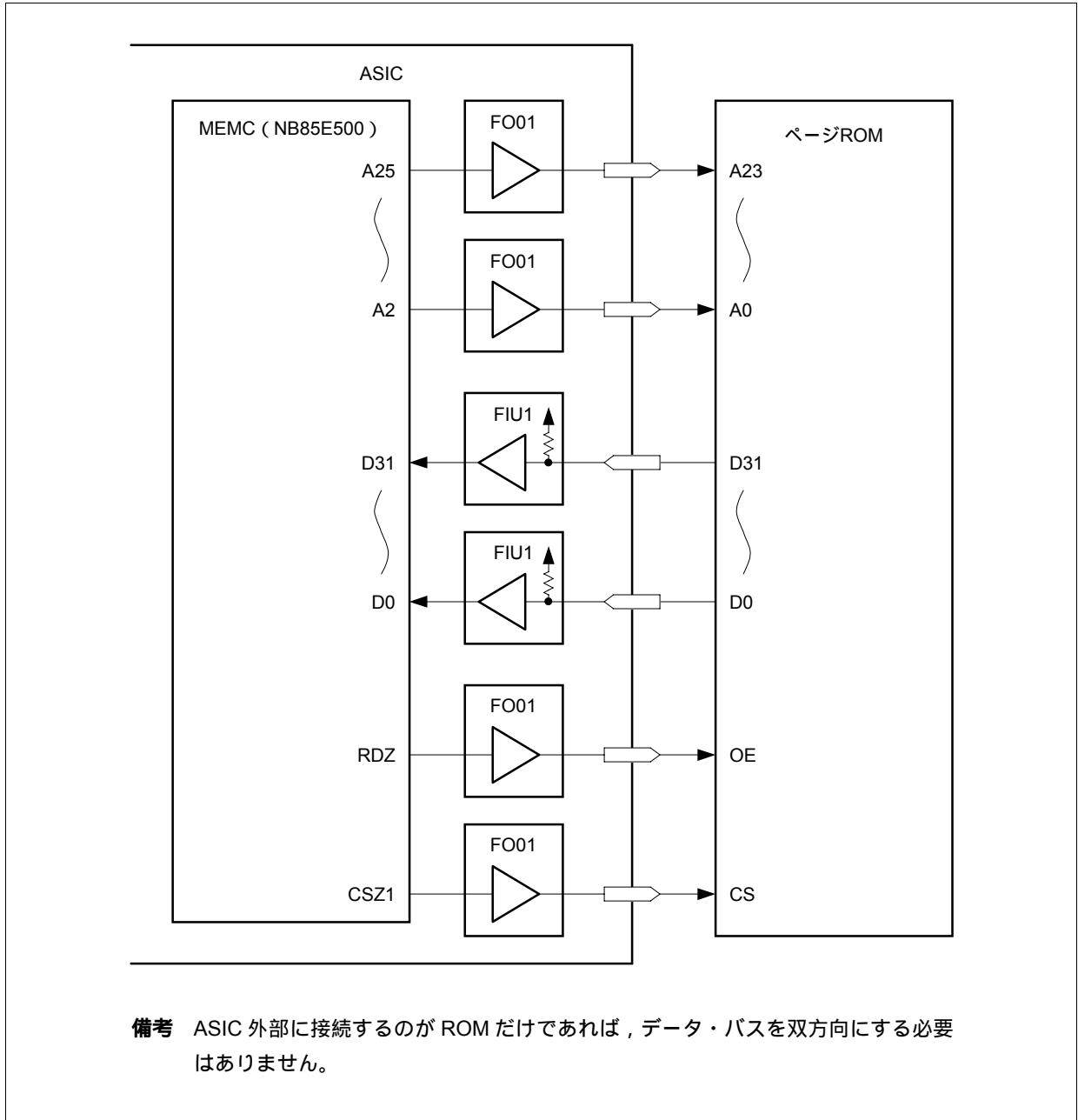
図5 - 2 SRAMとの接続例



5.2 ページROM との接続

32ビット幅のページROM との接続例を図5-3に示します。

図5-3 ページROMとの接続例



5.3 SDRAM との接続

MEMC (NU85E502) に SDRAM を接続する場合、次の条件に示す SDRAM である必要があります。

- リードのレイテンシ：2，または 3
- データ・バス幅：8 ビット，または 16 ビット，または 32 ビット
- ロウ・アドレス幅：11 ビット，または 12 ビット
- アドレス・マルチプレクス本数 (カラム・アドレス幅)：8 ビット，または 9 ビット，または 10 ビット
- リフレッシュ：CBR リフレッシュとセルフ・リフレッシュ

MEMC と SDRAM の接続例として 64M ビット (1M ワード × 16 ビット × 4 バンク構成) の SDRAM を 2 個接続し、32 ビット幅でアクセスする接続例を図 5 - 4 に示します。

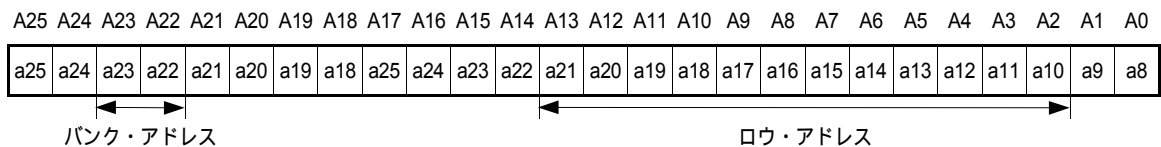
リフレッシュは、4096 サイクル / 64ms とします (64M ビットの SDRAM 相当品)。

この接続例の SDRAM は 64M ビット (1M ワード × 16 ビット × 4 バンク構成) で、リフレッシュ・サイクルは 4096 サイクルなので、次に示すアドレス構成になります。

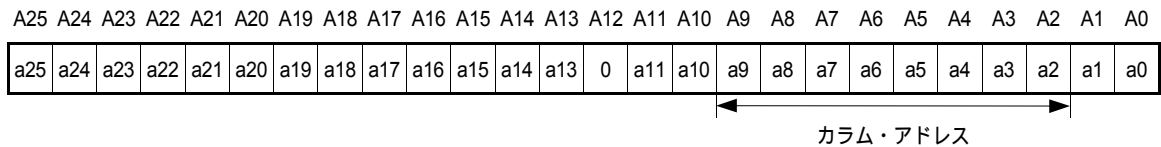
- ロウ・アドレス入力 : A0-A11 (12 本)
- カラム・アドレス入力 : A0-A7 (8 本)
- バンク・セレクト : A12, A13 (2 本)

また、32 ビット・データ・バス幅なのでアドレスを 2 ビット・シフトして NU85E502 と接続します。したがって、次のように NU85E502 と SDRAM のアドレスを接続します。

アクティブ・コマンド時のロウ・アドレス、バンク・アドレス出力



リード/ライト・コマンド時のカラム・アドレス出力

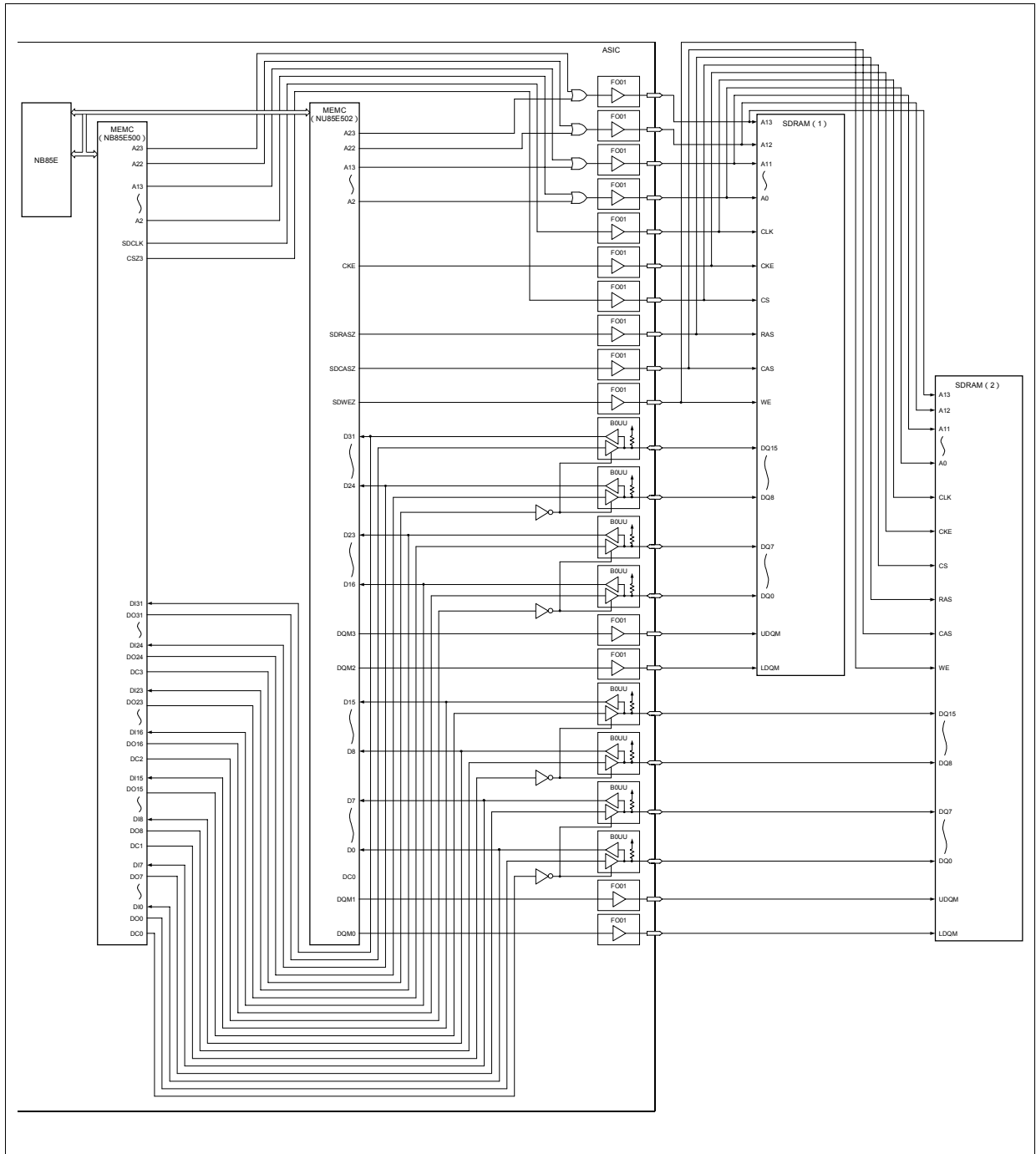


- NU85E502 と SDRAM の接続

A23, A22 (NU85E502) BA0 (A13), BA1 (A12) (SDRAM)

A13-A2 (NU85E502) A11-A0 (SDRAM)

図 5 - 4 SDRAMとの接続例



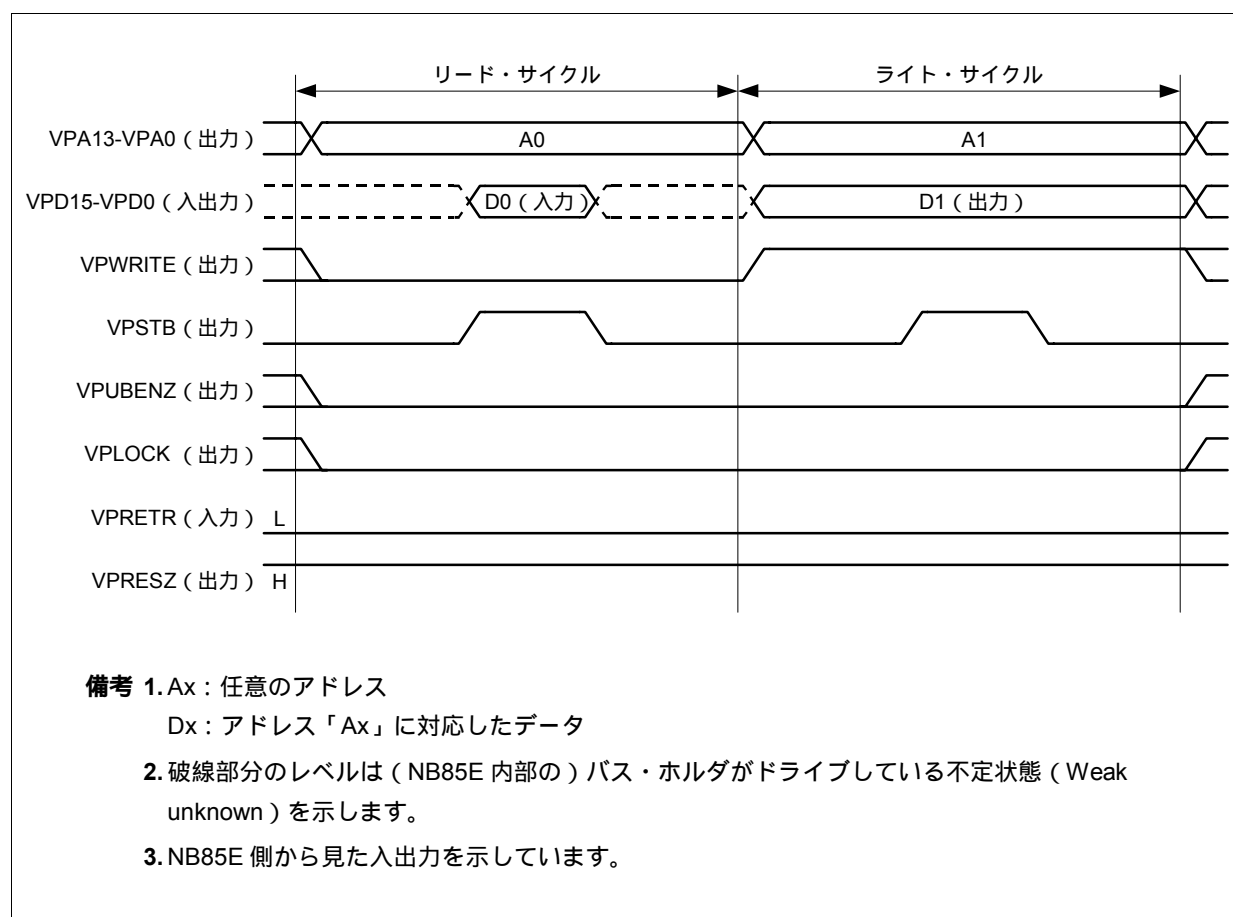
第 6 章 NPB への接続

6.1 NPB の概要

NB85E の NPB (NEC ペリフェラル・バス) は、バス・スピードが 10MHz 以下で動作する 16 ビット幅非同期バスです。10MHz 以下の周波数で動作するユーザ・ロジックが接続できます。

備考 MEMC, 命令キャッシュ, データ・キャッシュなど VSB に接続するマクロも NPB への接続が必要です。

図6 - 1 NPB動作タイミング・チャート



6.2 ユーザ・ロジックの接続

この節では、NPB にユーザ・ロジックを接続する例を示します。

ユーザ・ロジックの例として、8 ビットのレジスタを 2 個、16 ビットのレジスタを 1 個並べたものを作成します。ユーザ・ロジックのマクロ名を UDL1 とし、各レジスタ名を UDL11, UDL12, UDL13 とします。

(1) レジスタのマッピング

ユーザ・ロジックを配置するメモリ・アドレスは、プログラマブル周辺 I/O 領域か、次の範囲内の周辺 I/O 領域です（詳細は、**NB85E ユーザーズ・マニュアル ハードウェア編 (A13971J)** を参照してください）。

- xFFF200H-xFFF47FH
- xFFF520H-xFFF7BFH
- xFFF800H-xFFFFFFFH

ここでは、ユーザ・ロジックのレジスタを FFFFFFF880H 番地以降のアドレスに配置した例を示します。

表6-1 ユーザ・ロジックのアドレス割り当て例

アドレス	マクロ名	レジスタ名	ビット幅
FFFFFF880	UDL1	UDL11	8 ビット
FFFFFF881	UDL1	UDL12	8 ビット
FFFFFF882	UDL1	UDL13	16 ビット

(2) アドレス・デコーダの作成

ここでは、ユーザ・ロジックの接続例、アドレス・デコーダとユーザ・ロジックの HDL 作成例、タイミング・チャートを示します。

この例では、UDL11, UDL12, UDL13 とも 1 つのモジュールで処理するように設計されています。割り当てた番地数が 4 つなので、アドレス・バスは、VPA0, VPA1 の 2 本の信号で対応します。

注意 NPB にユーザ・ロジックを接続する場合、テスト・モード時（BUNRI 端子にアクティブ・レベルが入力されている場合）にデータ・バスがハイ・インピーダンスになるような設計を行ってください（10.2.1 NB85E, NB85ET の各端子の処理、NB85E ユーザーズ・マニュアル ハードウェア編 (A13971J) を参照してください）。

図6 - 2 ユーザ・ロジックの接続例

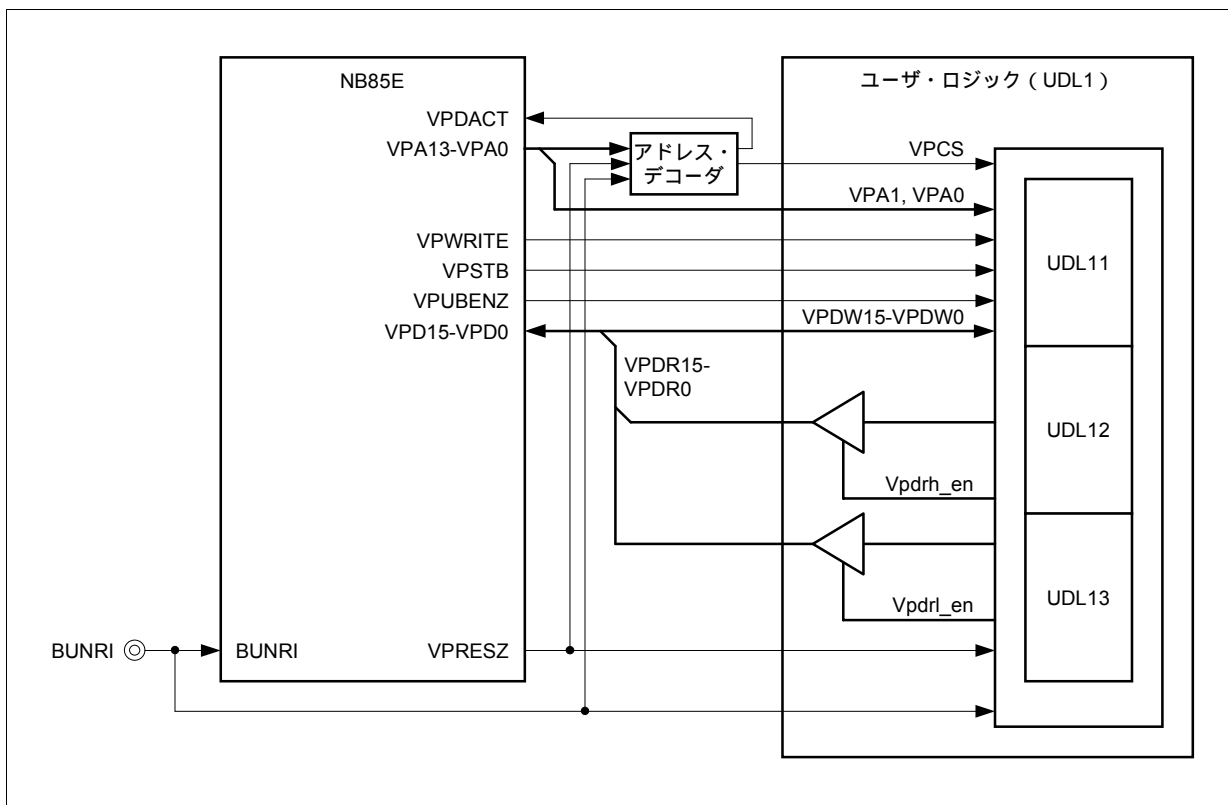


図6 - 3 アドレス・デコーダのHDL作成例

```

module ADRSDEC_UDL ( VPA, VPDACT, VPRESZ, UDL1_VPCS, BUNRI
);
parameter      UDL1_ADR      = 12'b11_1000_1000_00; // 3880H

input  [13:0]  VPA ;
input          VPRESZ, BUNRI ;

output        VPDACT ;
output        UDL1_VPCS ;

wire  [13:0]  VPA ;

assign VPDACT      = ~( UDL1_VPCS );
assign UDL1_VPCS   = ( UDL1_ADR == VPA[13:2] ) & ~BUNRI ;

endmodule

```

図6-4 ユーザ・ロジックのHDL作成例

```

module NPB_UDL1(
    VPA, VPDR, VPDW, VPCS, VPWRITE, VPSTB, VPUBENZ,
    VPRESZ, BUNRI
);

input  [1:0]  VPA ;
input  [15:0] VPDW ;
input  VPCS, VPWRITE, VPSTB, VPUBENZ, VPRESZ, BUNRI ;

output [15:0] VPDR ;

reg    [15:0] dout ;

//-- VPDR driver --

wire   vpdrl_en, vpdrh_en ;

assign vpdrl_en = VPCS & ~VPWRITE & VPSTB & ~VPA[0] & ~BUNRI ;
assign vpdrh_en = VPCS & ~VPWRITE & VPSTB & ~VPUBENZ & ~BUNRI ;

assign VPDR[7:0] = ( vpdrl_en ) ? dout[7:0] : 8'bzzzz_zzzz ;
assign VPDR[15:8] = ( vpdrh_en ) ? dout[15:8] : 8'bzzzz_zzzz ;

//-- User Logic Register --

reg    [7:0]  udl11, udl12 ;
reg    [15:0] udl13 ;

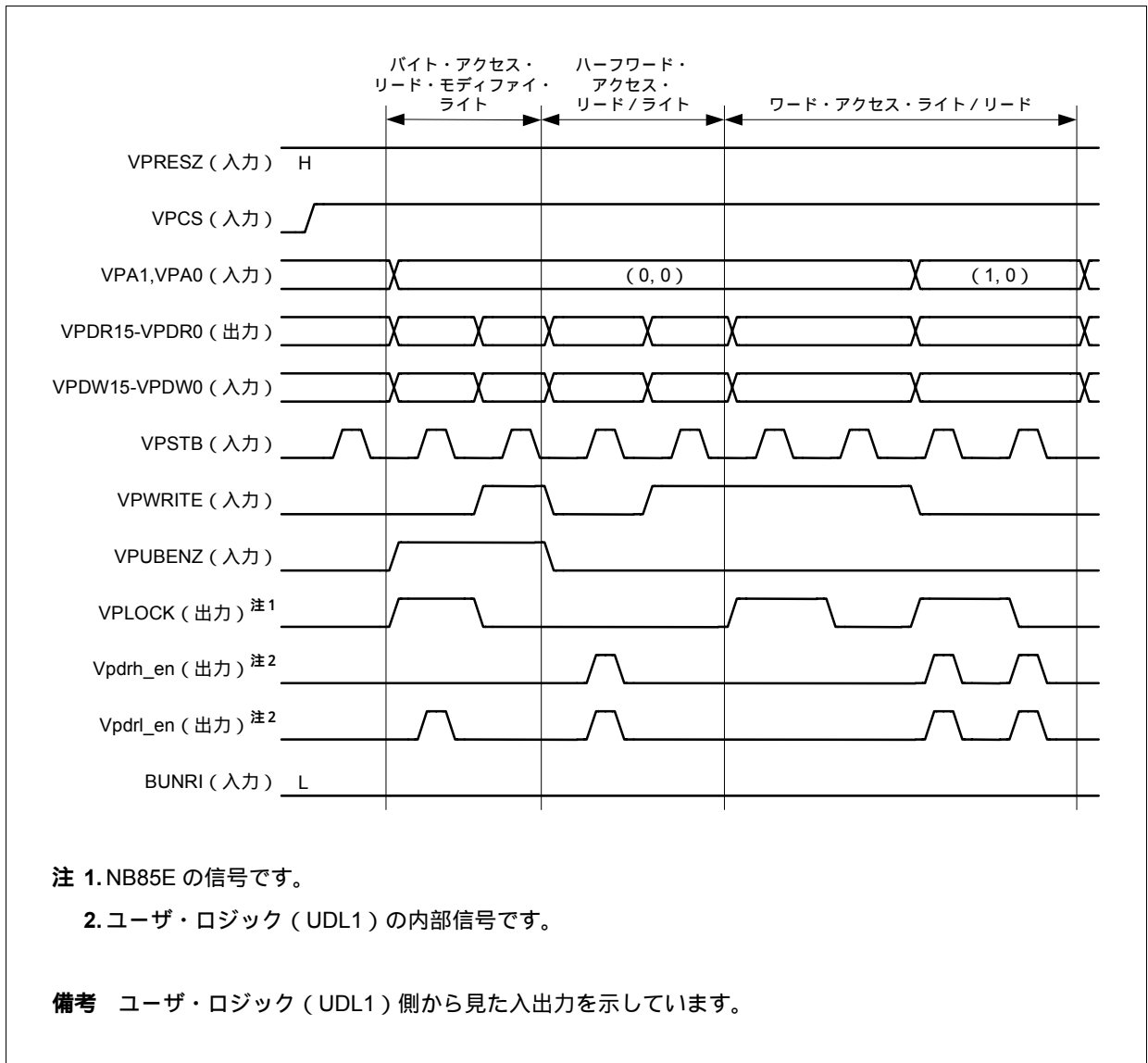
// synopsys async_set_reset "VPRESZ"

always @( VPSTB or VPRESZ ) begin
    if ( ~VPRESZ ) begin
        dout    <= 16'h0000 ;
        udl11   <= 8'h00 ;
        udl12   <= 8'h00 ;
        udl13   <= 16'h0000 ;
    end
    else begin
        if ( VPSTB & VPCS ) begin
            if ( ~VPA[1] ) begin
                if ( ~VPWRITE ) begin // read action
                    dout <= { udl12, udl11 } ;
                end
                else begin // write action
                    if ( VPA[0] & ~VPUBENZ ) udl12 <= VPDW[15:8] ;
                    else if ( ~VPA[0] & VPUBENZ ) udl11 <= VPDW[7:0] ;
                    else if ( ~VPA[0] & ~VPUBENZ ) begin
                        udl12 <= VPDW[15:8] ;
                        udl11 <= VPDW[7:0] ;
                    end
                end
            end
        end
        else begin
            if ( ~VPWRITE ) dout <= udl13 ;
            else udl13 <= VPDW ;
        end
    end
end

endmodule

```

図6-5 ユーザ・ロジックの動作タイミング・チャート



(3) 応用例 (リトライ機能付きのユーザ・ロジックの作成例)

上記の設計例で示したユーザ・ロジックを非常に低速なマクロと NPB との仲介用のバッファとして使用する場合、ユーザ・ロジックにリトライ機能を付けることによって実現できます。

リトライ機能付きユーザ・ロジックの HDL 作成例を図 6-6 に、動作タイミング・チャートを図 6-7 に示します。

なお、この例では、UDL13 を読み出す場合に、VPRETR 信号 1 回分のウェイトが入るように設計されています。

図6 - 6 リトライ機能付きユーザ・ロジックのHDL作成例 (1/2)

```

module NPB_UDL2(
    VPA, VPDR, VPDW, VPCS, VPWRITE, VPSTB, VPUBENZ,
    VPRESZ, VPRETR, BUNRI
);

input  [1:0]  VPA ;
input  [15:0] VPDW ;
input   VPCS, VPWRITE, VPSTB, VPUBENZ, VPRESZ, BUNRI ;

output [15:0] VPDR ;
output   VPRETR ;

reg    [15:0] dout ;
reg    VPRETR ;

//-- VPDR driver --

wire   vpdrl_en, vpdrh_en ;

assign vpdrl_en = VPCS & ~VPWRITE & VPSTB & ~VPA[0] & ~BUNRI ;
assign vpdrh_en = VPCS & ~VPWRITE & VPSTB & ~VPUBENZ & ~BUNRI ;

assign VPDR[7:0] = ( vpdrl_en ) ? dout[7:0] : 8'bzzzz_zzzz ;
assign VPDR[15:8] = ( vpdrh_en ) ? dout[15:8] : 8'bzzzz_zzzz ;

//-- User Logic Register --

reg    [7:0]  udl11, udl12 ;
reg    [15:0] udl13 ;

reg    retract ;

// synopsys async_set_reset "VPRESZ"

always @( VPSTB or VPRESZ ) begin
    if ( ~VPRESZ ) begin
        dout    <= 16'h0000 ;
        udl11   <= 8'h00 ;
        udl12   <= 8'h00 ;
        udl13   <= 16'h0000 ;
        VPRETR <= 1'b0 ;
    end
    else begin
        if ( VPSTB & VPCS ) begin
            if ( ~VPA[1] ) begin
                if ( ~VPWRITE ) begin // read action
                    dout <= { udl12, udl11 } ;
                end
                else begin // write action
                    if ( VPA[0] & ~VPUBENZ ) udl12 <= VPDW[15:8] ;
                    else if ( ~VPA[0] & VPUBENZ ) udl11 <= VPDW[7:0] ;
                    else if ( ~VPA[0] & ~VPUBENZ ) begin
                        udl12 <= VPDW[15:8] ;
                        udl11 <= VPDW[7:0] ;
                    end
                end
            end
        end
        else begin
            if ( ~VPWRITE ) begin
                VPRETR <= retract ;
                dout    <= udl13 ;
            end
            else udl13 <= VPDW ;
        end
    end
end

end
end

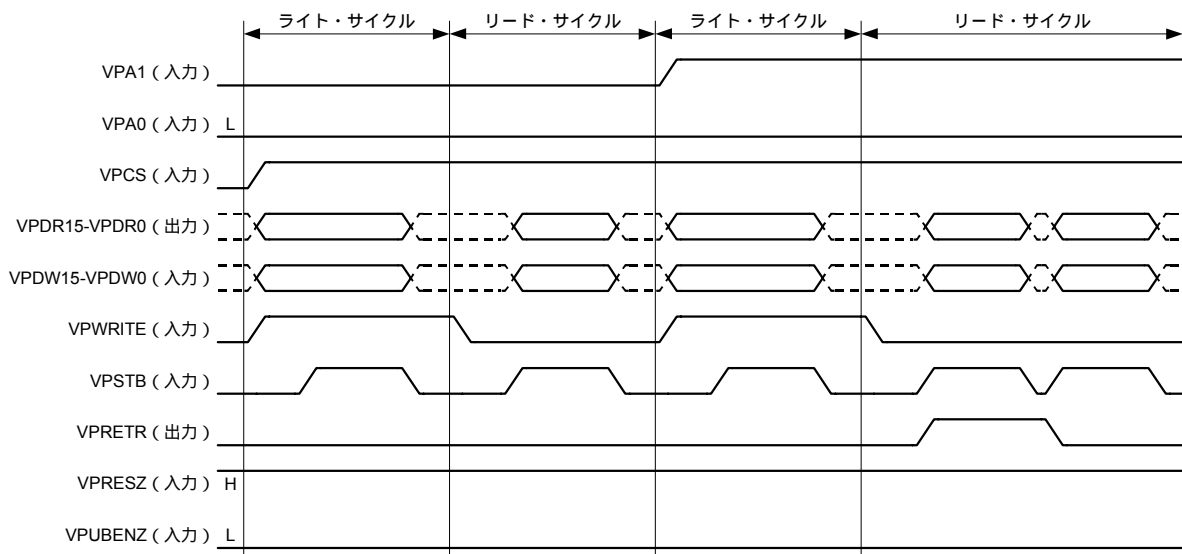
```

図6 - 6 リトライ機能付きユーザ・ロジックのHDL作成例 (2/2)

```

always @(posedge VPSTB or negedge VPRESZ) begin
    if (~VPRESZ) retract <= 1'b0;
    else if (~VPWRITE & VPCS & VPA[1]) retract <= ~retract;
end
endmodule
    
```

図6 - 7 リトライ機能付きユーザ・ロジックの動作タイミング・チャート



- 備考** 1.破線部分のレベルは (NB85E 内部の) バス・ホルダがドライブしている不定状態 (Weak unknown) を示します。
- 2.ユーザ・ロジック (UDL1) 側から見た入出力を示しています。

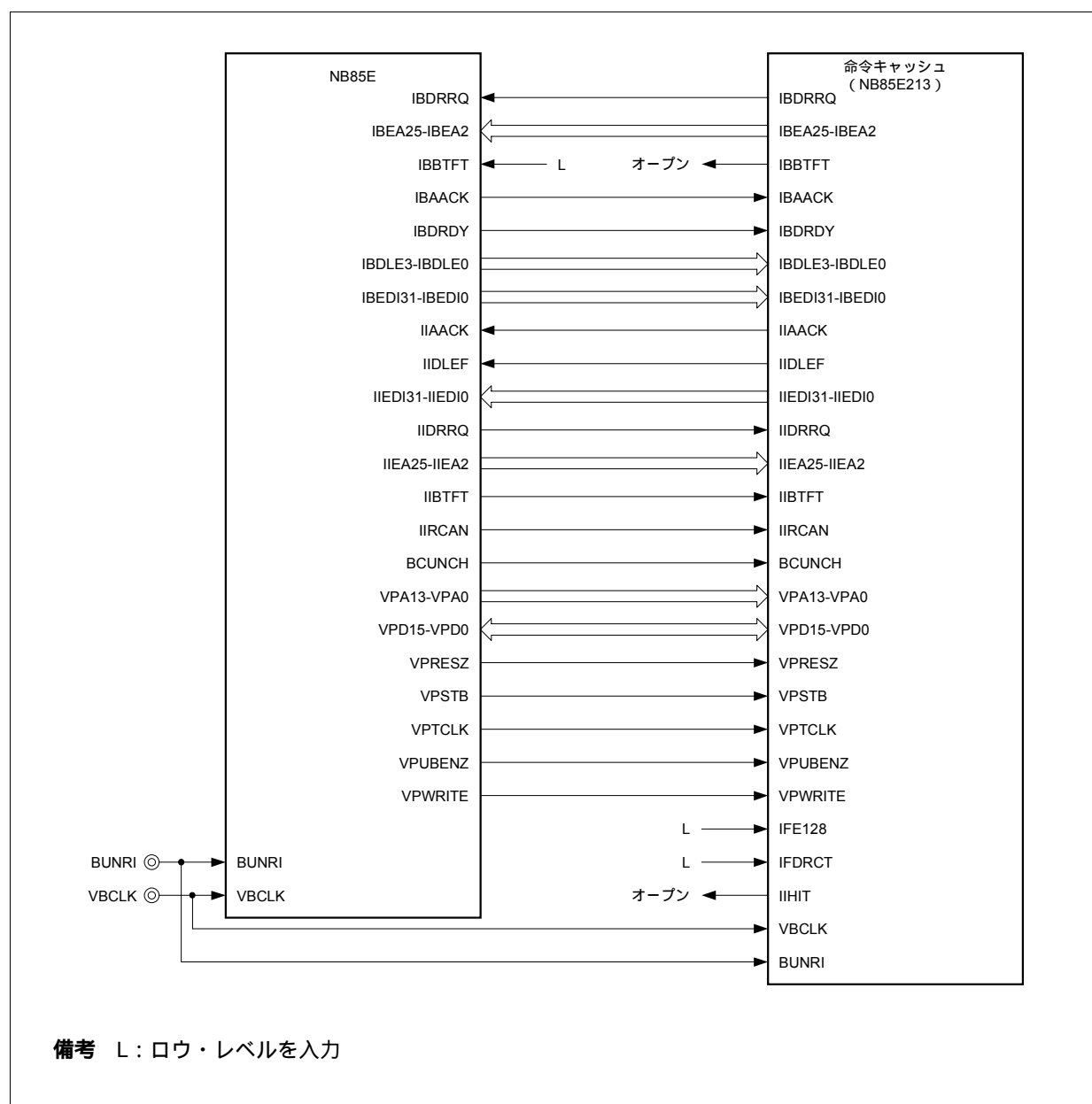
第7章 キャッシュの接続

7.1 命令キャッシュの接続

NB85E に接続可能な命令キャッシュには、次の製品があります。

- NB85E212 ... 4K バイト 2 ウエイ・セット・アソシアティブ命令キャッシュ
(4ワード × 128 エントリ × 2 ウエイ = 4K バイト)
- NB85E213 ... 8K バイト 2 ウエイ・セット・アソシアティブ命令キャッシュ
(4ワード × 256 エントリ × 2 ウエイ = 8K バイト)

図7-1 NB85Eと命令キャッシュ (NB85E213) の接続例



7.1.1 命令キャッシュ使用時の注意事項

(1) NB85E との接続

同じ端子名の端子同士を接続してください。ただし、命令キャッシュの IBTFT 端子はオープンにし、NB85E の IBTFT 端子はロウ・レベルに固定してください。

(2) キャッシュ・タイプ選択用端子の設定

IF で始まるキャッシュ・タイプ選択用端子には次に示すレベルを入力してください。

端子名称	入力レベル	
	NB85E212	NB85E213
IFE128	ハイ・レベル	ロウ・レベル
IFDRCT	ロウ・レベル	ロウ・レベル

(3) バス・サイクルの状態

NB85E のキャッシュ・コンフィギュレーション・レジスタ (BHC) により命令キャッシュの設定をキャッシュ可能に設定した領域では、NB85E の VBCTYP2-VBCTYP0 信号は常に通常のおペコード・フェッチを示し、分岐命令による飛び先アドレスのおペコード・フェッチは示しません。

(4) リセット時の動作

リセット時には、自動的にタグをクリア (無効に) し、次のデータの置き換えはウエイ 0 から行われる状態にします。このため、リセット後ライン数分のクロック・サイクルの期間に命令キャッシュへのアクセスがあると、CPU はタグがクリアされる (無効になる) まで停止します。

(5) レジスタの設定

次に示す NB85E のレジスタの設定は必ずキャッシュ不可領域で行ってください。ただし、命令キャッシュ・コントロール・レジスタ (ICC) のビット 4 の設定はキャッシュ可能領域で行ってください。

- チップ領域セレクト制御レジスタ (CSC0, CSC1)
- 周辺 I/O 領域セレクト制御レジスタ (BPC)
- バス・サイズ・コンフィギュレーション・レジスタ (BSC)
- エンディアン・コンフィギュレーション・レジスタ (BEC)
- キャッシュ・コンフィギュレーション・レジスタ (BHC)
- 命令キャッシュ・コントロール・レジスタ (ICC)[※]
- 命令キャッシュ・データ・コンフィギュレーション・レジスタ (ICD)

注 ビット 4 を除く

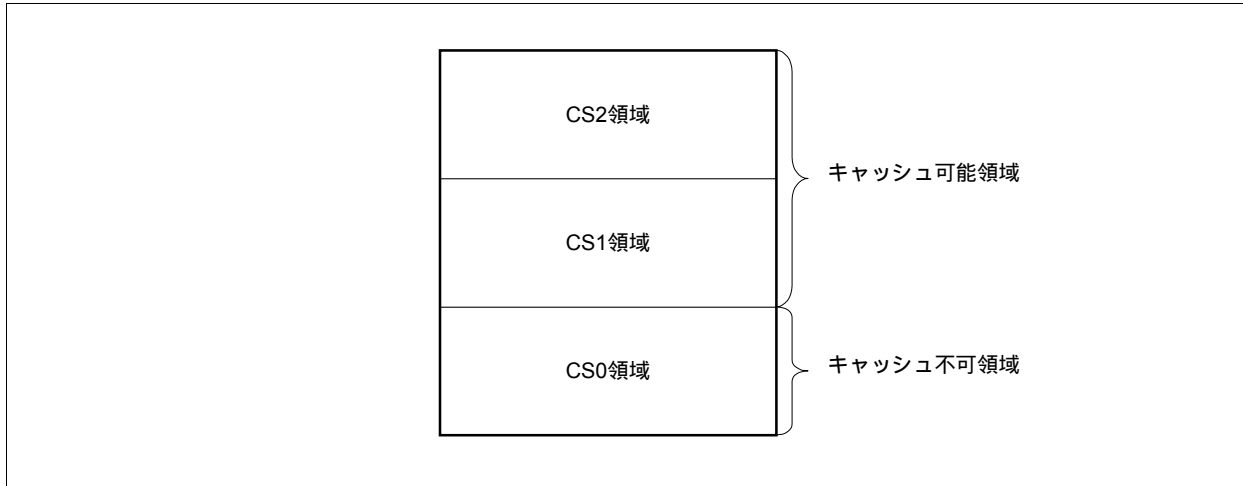
(6) メモリ境界へのアクセス

隣接するチップ・セレクト (CSn) 領域がキャッシュ可能領域とキャッシュ不可領域であった場合、このメモリ境界間を連続的にアクセスする場合は、分岐命令によってだけアクセス可能です (n = 7-0)。分岐命令以外でこのメモリ境界を連続アクセスした場合の動作は保証しません。次に例を示します。

例 キャッシュ領域の設定が図7-2のようになっています。この場合、メモリ領域へのアクセスは次のようになります。

- CS0 領域から CS1 領域へは分岐命令でだけアクセス可能
- CS1 領域から CS2 領域へは連続的にアクセス可能

図7-2 キャッシュ領域設定例



(7) プログラムの初期設定

システム・リセット直後のユーザ・プログラムの初期設定で、NB85E の BHC レジスタを設定する前に、必ず次の命令を実行してください。

```
st.h    r0,0xffff072[r0]
```

この命令を実行したあとに、BHC レジスタにより命令キャッシュの設定をキャッシュ可能 (BHN0 ビット = 1) にすることでキャッシュが有効となります (n = 7-0)。

(8) NB85E の BHC レジスタの設定

BHC レジスタを設定する命令が存在する CSn 領域に対しては、その命令による命令キャッシュのキャッシュ可能 / 不可の設定ができません (n = 7-0)。BHC レジスタを設定する命令が存在しない CSn 領域に対してのみ、命令キャッシュのキャッシュ可能 / 不可の設定ができます。

たとえば、CS0 領域に BHC レジスタ設定命令が存在するときは、CS0 領域の命令キャッシュの設定 (キャッシュ可能 / 不可の設定) はできません。この場合は、CS1-CS7 領域のみ、命令キャッシュの設定ができます。

ただし、VFB または VDB に接続されたメモリ領域にある命令からはすべての CSn 領域に対して命令キャッシュの設定が可能です。

備考 VFB : ROM 直結のための専用バス (V850E フェッチ・バス)

VDB : RAM 直結のための専用バス (V850E データ・バス)

(9) テスト・バス自動結線ツールの対応

この命令キャッシュには BUNRI 端子がありますが、テスト・バス (TBOx, TBix) がないため、テスト・バス自動結線ツールには対応していません。

(10) タグ・クリア手順

ウエイ 0 とウエイ 1 のタグ・クリア用のカウンタは共用されています。

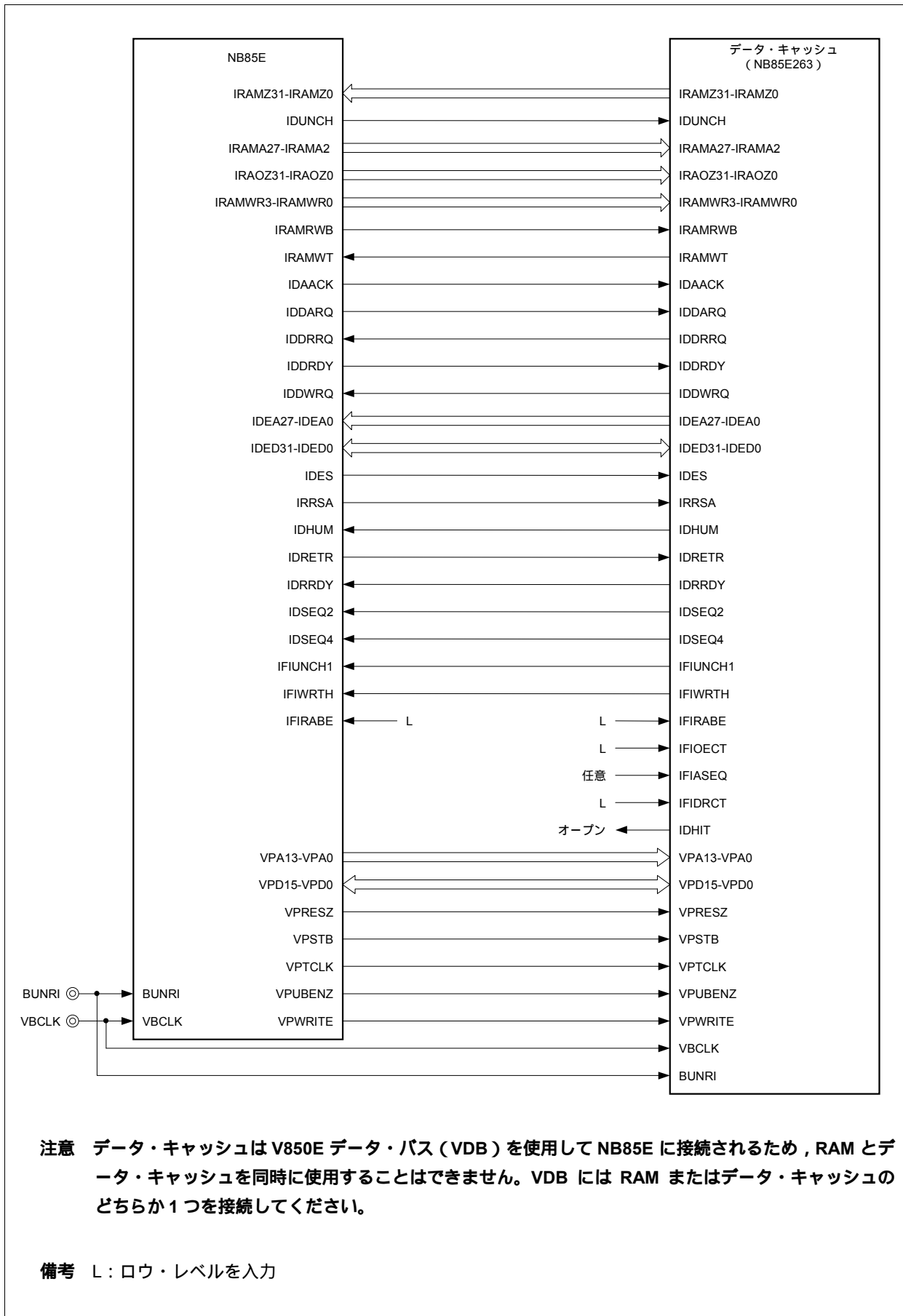
したがって、タグ・クリアの実行 (ICC レジスタの TCLR0 ビット、または TCLR1 ビットのセット (1)) は、タグ・クリア用のカウンタ停止時 (TCLR0 = TCLR1 = 0) に行ってください。ウエイ 0 とウエイ 1 のタグ・クリアを別々に行う場合、片方のウエイのタグ・クリア実行中 (TCLR0 または TCLR1 = 1) に、もう一方のウエイのタグ・クリアを実行すると、タグ・クリアの途中でカウンタが止まります。そのため、カウンタが途中の値を示したまま、もう一方のウエイのタグ・クリア動作に移行してしまうため、正常なタグ・クリアが行えません。必ず一方のウエイのタグ・クリアが完了したことを確認してから (TCLR0 または TCLR1 = 0)、もう一方のウエイのタグ・クリアを行うようにしてください。

7.2 データ・キャッシュの接続

NB85E に接続可能なデータ・キャッシュには、次の製品があります。

- NB85E252 ... 4K バイト・ダイレクト・マップ・データ・キャッシュ
(4 ワード × 256 エントリ = 4K バイト)
- NB85E263 ... 8K バイト 2 ウエイ・セット・アソシアティブ・データ・キャッシュ
(4 ワード × 256 エントリ × 2 ウエイ = 8K バイト)

図7-3 NB85Eとデータ・キャッシュ (NB85E263) の接続例



7.2.1 データ・キャッシュ使用時の注意事項

(1) NB85E との接続

同じ端子名の端子同士を接続してください。

(2) キャッシュ・タイプ選択用端子の設定

IFI で始まるキャッシュ・タイプ選択用端子には次に示すレベルを入力してください。

なお、IFIUNCH1, IFIWRTH 端子は NB85E に接続してください。

端子名称	入力レベル	
	NB85E252	NB85E263
IFIASEQ	任意 ^{注1}	任意 ^{注1}
IFIRABE	ロウ・レベル	ロウ・レベル
IFIDRCT ^{注2}	—	ロウ・レベル
IFIOECT	ロウ・レベル	ロウ・レベル

注 1. IFIASEQ は、システムに応じて設定してください。詳細については、**命令キャッシュ、データ・キャッシュ ユーザーズ・マニュアル NB85E, NB85ET 編 (A14247J)** を参照してください。

2. NB85E263 のみ。

(3) バス・サイクルの状態

NB85E のキャッシュ・コンフィギュレーション・レジスタ (BHC) により、データ・キャッシュの設定をキャッシュ可能に設定した領域のすべてのリード・サイクル、およびライトバック・モード (ライト・アロケート許可) 時のライト・サイクルでは、NB85E の VBCTYP2-VBCTYP0 信号は常にデータ・アクセスを示し、ミス・アライン・アクセスは示しません。

(4) リセット時の動作

リセット時には、自動的にタグをクリア (無効に) し、次のデータの置き換えはウエイ 0 から行われる状態にします。このため、リセット後ライン数分のクロック・サイクルの期間にデータ・キャッシュへのアクセスがあると、CPU はタグがクリアされる (無効になる) まで停止します。

(5) テスト・バス自動結線ツールの対応

このデータ・キャッシュには BUNRI 端子がありますが、テスト・バス (TBOx, TBix) がないため、テスト・バス自動結線ツールには対応していません。

(6) その他

バス・スヌープ回路（バスの動作を監視する回路）は内蔵されていません。次の例に示すような場合、データ・キャッシュへのライト・アクセスがなくてもデータ・キャッシュ内のデータはダーティ・データとなり、コヒーレンシ性が崩れてしまいます。したがって、この状態を回避するために、必ずタグ・クリアを行ってください。

例 1. キャッシュ可能領域の外部メモリに対して DMA 転送を行った場合

（データ・キャッシュに転送データは反映されません）

2. 外部バス・マスタがキャッシュ可能領域の外部メモリ内容を書き換えた場合など**(7) デバッグ中の動作**

N-Wire 型インサーキット・エミュレータによるデバッグ中は、データ・キャッシュは動作しません。

デバッグ中にキャッシュ許可領域の外部メモリをアクセスする場合は、データ・キャッシュが有効であっても直接外部メモリのみにアクセスするため、コヒーレンシ性が崩れてしまいます。これを回避するために必ずデータ・キャッシュのタグ・クリアを行ってください。また、インサーキット・エミュレータ（IE-V850E-MC-A）を使用する場合は、データ・キャッシュのデバッグは行えません。

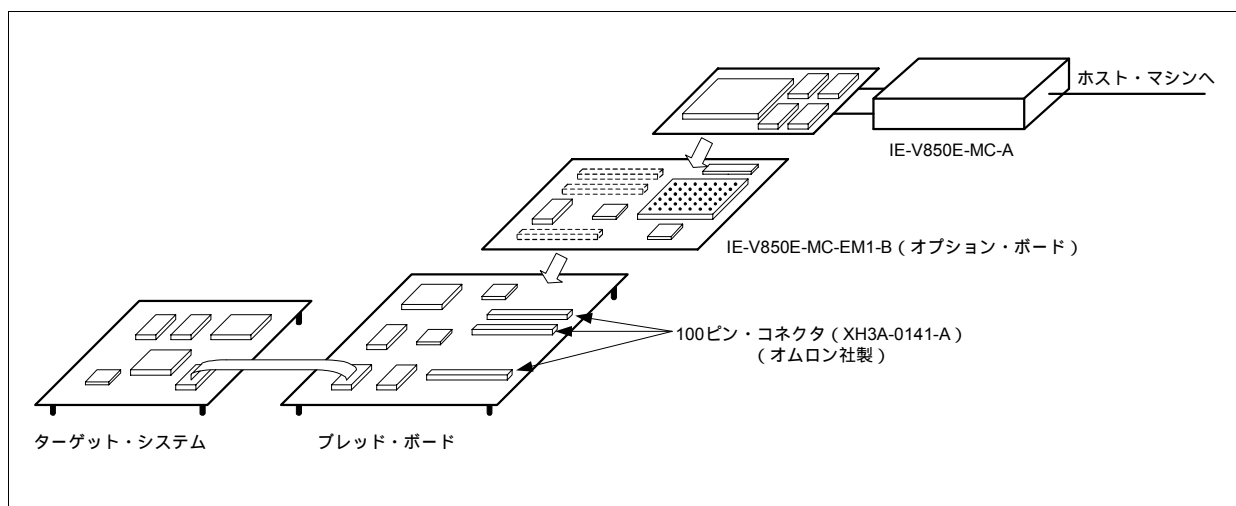
第 8 章 インサーキット・エミュレータ (IE) との接続

8.1 NB85E 用インサーキット・エミュレータ (IE-V850E-MC-A) を使用する場合

オプション・ボード (IE-V850E-MC-EM1-B) と接続するために、ブレッド・ボード上に 3 個の 100 ピン・コネクタ[※]を実装する必要があります (コネクタの配置, 寸法や信号一覧については, IE-V850E-MC-EM1-B, IE-V850E-MC-MM2 ユーザーズ・マニュアル (U14482J) を参照してください)。

注 推奨コネクタ : XH3A-0141-A (オムロン社製)

図8 - 1 NB85E用インサーキット・エミュレータ (IE-V850E-MC-A) の接続



注意 1. IE-V850E-MC-A を使用したディバグには, 命令/データ・キャッシュ用端子, テスト・モード用端子, VFB/VDB 用端子, RCU 用端子, 周辺エパチップ・モード用端子は不要なため, これらの端子は, オプション・ボードにはありません。

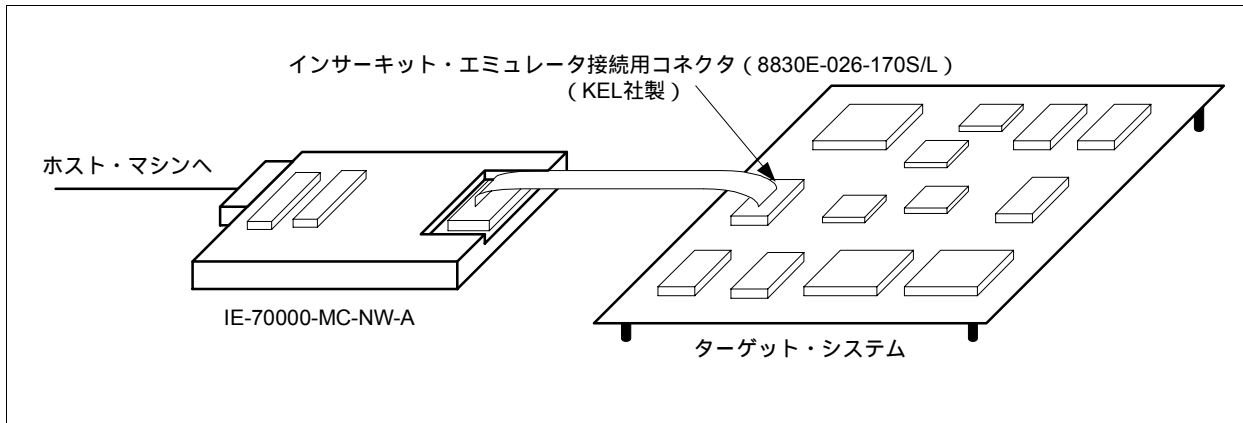
また, 動作モード設定用端子もオプション・ボードにはありませんが, IE-V850E-MC-A の CPU 制御レジスタにより動作モードの初期値が設定できます。

2. ディバグを行う際は, インサーキット・エミュレータ本体 (IE-V850E-MC-A) とオプション・ボード (IE-V850E-MC-EM1-B) に添付の制限文書を必ずお読みください。

8.2 N-Wire 型インサーキット・エミュレータ (IE-70000-MC-NW-A) を使用する場合

ターゲット・システム上にインサーキット・エミュレータ接続用コネクタと接続用回路を実装する必要があります。

図8-2 N-Wire型インサーキット・エミュレータの接続



8.2.1 インサーキット・エミュレータ接続コネクタ (ターゲット・システム側)

図8-3にインサーキット・エミュレータ接続コネクタ (ターゲット・システム側) のピン配置図を、表8-1にピン機能を示します。

備考 推奨コネクタは次のとおりです。

- 8830E-026-170S (KEL 社製) : 26 ピン・ストレート・タイプ
- 8830E-026-170L (KEL 社製) : 26 ピン・ライト・アングル・タイプ

図8-3 インサーキット・エミュレータ接続コネクタ (ターゲット・システム側) のピン配置図

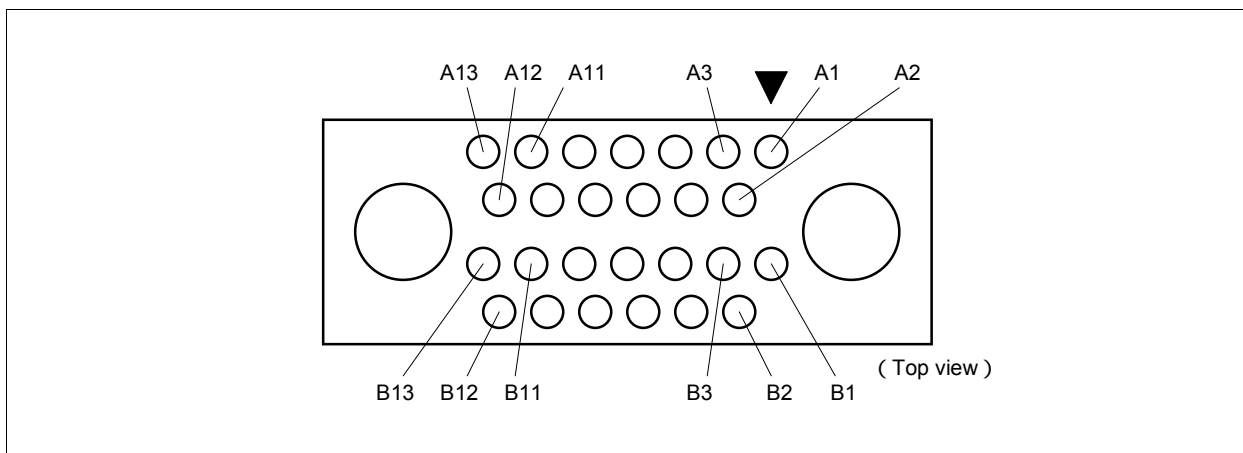


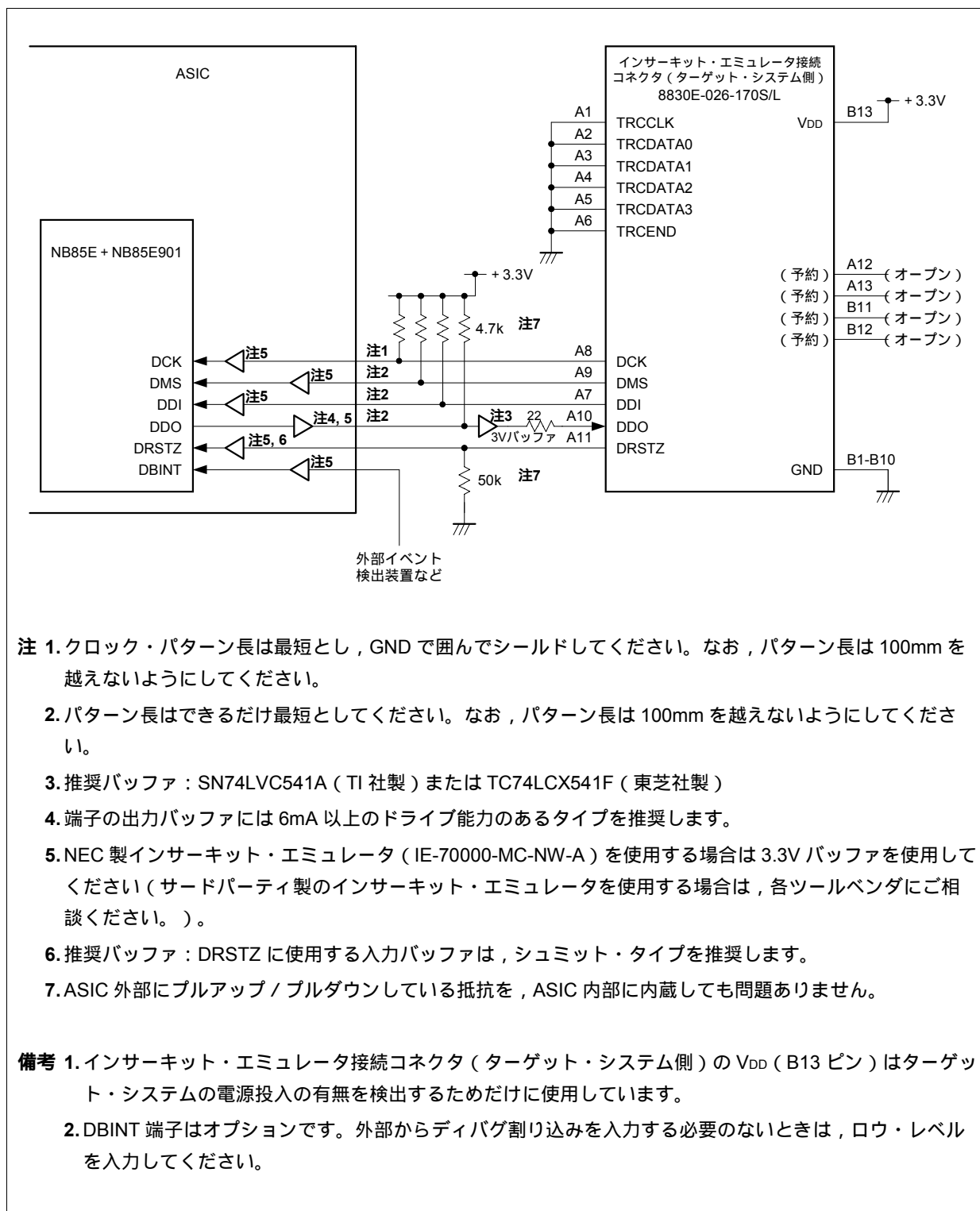
表8-1 インサーキット・エミュレータ接続コネクタ(ターゲット・システム側)ピン機能

ピン番号	端子名	入出力	端子機能
A1	TRCCLK	入力	トレース・クロック入力
A2	TRCDATA0	入力	トレース・データ0入力
A3	TRCDATA1	入力	トレース・データ1入力
A4	TRCDATA2	入力	トレース・データ2入力
A5	TRCDATA3	入力	トレース・データ3入力
A6	TRCEND	入力	トレース・データ・エンド入力
A7	DDI	出力	ディバグ・シリアル・インタフェース用データ出力
A8	DCK	出力	ディバグ・シリアル・インタフェース用クロック出力
A9	DMS	出力	ディバグ・シリアル・インタフェース用転送モード選択出力
A10	DDO	入力	ディバグ・シリアル・インタフェース用データ入力
A11	DRSTZ	出力	DCU リセット出力
A12	(予約)	-	(オープンにしてください)
A13	(予約)	-	(オープンにしてください)
B1	GND	-	-
B2	GND	-	-
B3	GND	-	-
B4	GND	-	-
B5	GND	-	-
B6	GND	-	-
B7	GND	-	-
B8	GND	-	-
B9	GND	-	-
B10	GND	-	-
B11	(予約)	-	(オープンにしてください)
B12	(予約)	-	(オープンにしてください)
B13	VDD	-	+3.3V 入力(ターゲット電源投入監視用)

8.2.2 NB85E に RCU (NB85E901) を接続した場合の回路例

図8-4 にインサーキット・エミュレータ接続コネクタ部 (ターゲット・システム側) の回路例を示します。

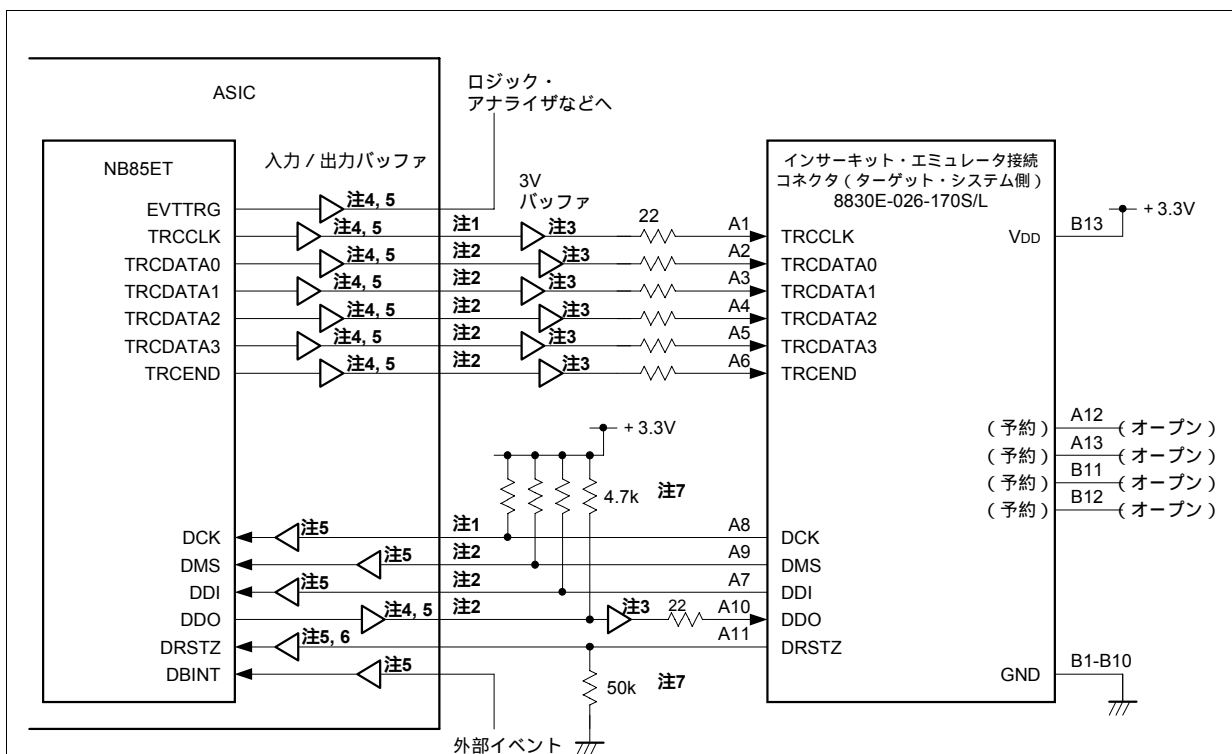
図8-4 インサーキット・エミュレータ接続回路例 (NB85E + RCU (NB85E901))



8.2.3 NB85ET を接続した場合の回路例

図 8 - 5 にインサーキット・エミュレータ接続コネクタ部 (ターゲット・システム側) の回路例を示します。

図8 - 5 インサーキット・エミュレータ接続回路例 (NB85ET)



- 注 1. クロック・パターン長は最短とし, GND で囲んでシールドしてください。なお, パターン長は 100mm を越えないようにしてください。
2. パターン長はできるだけ最短としてください。なお, パターン長は 100mm を越えないようにしてください。
3. 推奨バッファ: SN74LVC541A (TI 社製) または TC74LCX541F (東芝社製)
4. 端子の出力バッファには 6mA 以上のドライブ能力のあるタイプを推奨します。
5. NEC 製インサーキット・エミュレータ (IE-70000-MC-NW-A) を使用する場合は 3.3V バッファを使用してください (サードパーティ製のインサーキット・エミュレータを使用する場合は, 各ツールベンダにご相談ください。)。
6. 推奨バッファ: DRSTZ に使用する入力バッファは, シュミット・タイプを推奨します。
7. ASIC 外部にプルアップ / プルダウンしている抵抗を, ASIC 内部に内蔵しても問題ありません。

- 備考 1. インサーキット・エミュレータ接続コネクタ (ターゲット・システム側) の V_{DD} (B13 ピン) はターゲット・システムの電源投入の有無を検出するためだけに使用しています。
2. DBINT 端子はオプションです。外部からデバッグ割り込みを入力する必要のないときは, ロウ・レベルを入力してください。
3. EVTTRG 端子はオプションです。主にロジック・アナライザなどの計測装置のトリガ出力として使用できます。トリガ出力の必要がないときは, オープンにしてください。

8.2.4 N-Wire 型インサーキット・エミュレータとのタイミング設計

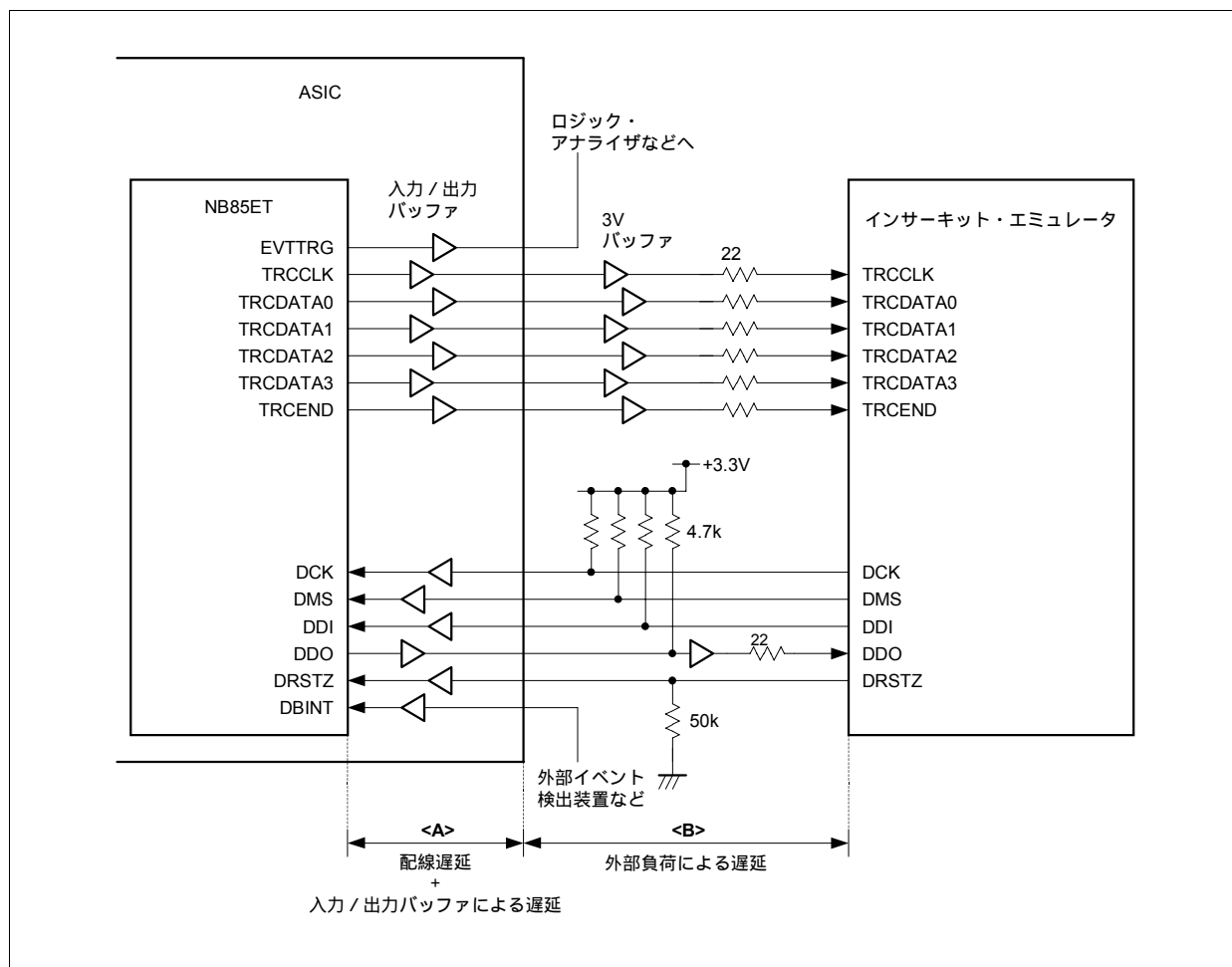
NB85ET (または NB85E901) と N-Wire 型インサーキット・エミュレータ間の信号のタイミング設計については、次に示す 2 つの遅延を考慮して NB85ET (または NB85E901) , および N-Wire 型インサーキット・エミュレータのタイミング・スペックを満足するように設計してください。

- 配線遅延 + 出力バッファによる遅延
(NB85ET (または NB85E901) のマクロから ASIC の端子間 (図 8 - 6 の<A>参照))
- 外部負荷による遅延
(ASIC の端子から N-Wire 型インサーキット・エミュレータ間 (図 8 - 6 の参照))

備考 NB85ET (または NB85E901) のタイミング・スペックについては **CB-9 ファミリ VX/VM タイプ 設計マニュアル CPU コア, メモリ・コントローラ編 (A13195J)** を参照してください。N-Wire 型インサーキット・エミュレータ端子のスペックについて、IE-70000-MC-NW-A を使用する場合は NEC へ、NEC 製品以外の N-Wire 型インサーキット・エミュレータを使用する場合は各サードパーティ・ツール・ベンダにご相談ください。

図 8 - 6 に NB85ET と N-Wire 型インサーキット・エミュレータ間の遅延について示します。

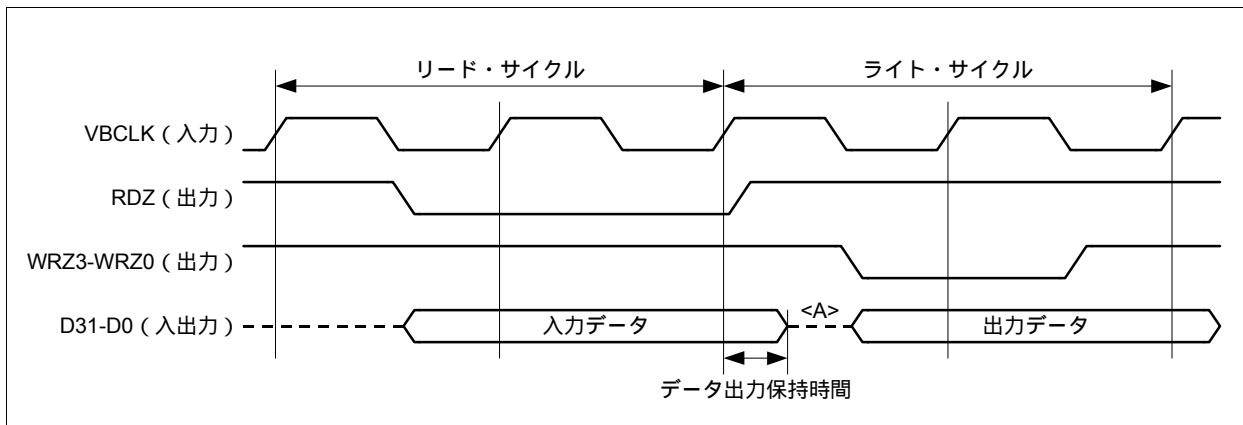
図8 - 6 NB85ETとN-Wire型インサーキット・エミュレータ間の遅延



第9章 注意事項

9.1 リード・サイクルからライト・サイクルに移行する際のバス・コンテンション (MEMC (NB85E500) 使用時)

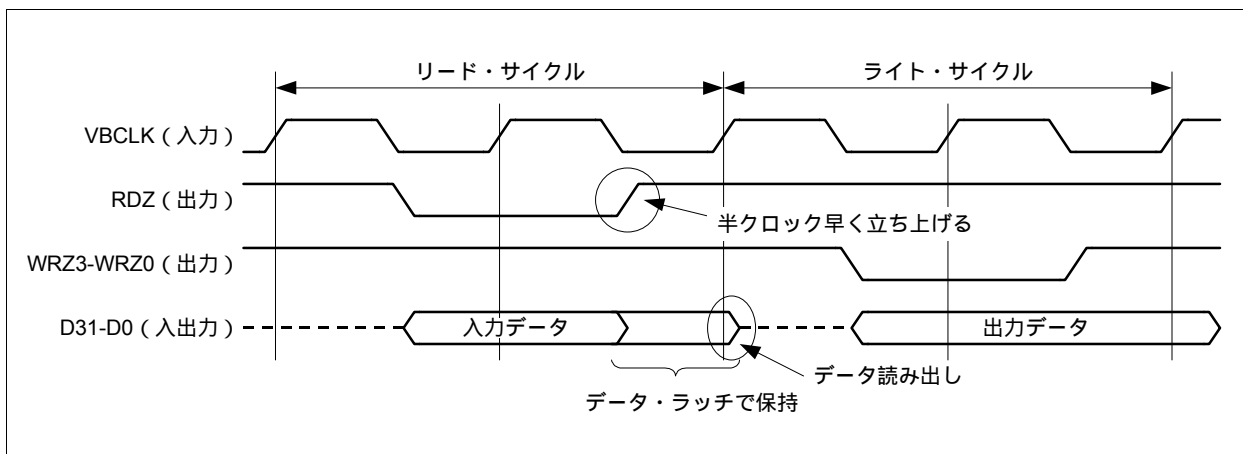
リード・サイクルのあとにライト・サイクルが発生するような場合 (外部 ROM から読み込んだデータを外部 RAM に書き込む場合など), 通常, 次のようなバス・タイミングとなります。



このとき, 外部 ROM のデータ出力保持時間が VbCLK の半クロック以上長くなると, 外部 ROM からの入力データと外部 RAM への出力データが衝突する可能性があります (上図の<A>)。

回避策として, 次の2種類の対策があります。

- (1) バス・サイクル・コントロール・レジスタ (BCC) の設定により, アイドル・サイクルを挿入します。
リード・サイクルの後ろにアイドル・サイクルが挿入されるように設定してください。ただし, ROM アクセス (リード・サイクル) のパフォーマンスは低下します。
- (2) リード・ストローブ信号 (RDZ) を半クロック早く立ち上げます。ただし, NB85E のデータ読み出しはリード・サイクルの最後のクロックの立ち上がりなので, データの消滅を防ぐためにチップ内部にデータ・ラッチを追加する必要があります。



9.2 Verilog シミュレーションに関する注意

NB85E の Verilog シミュレーション・モデルでは、「MOV imm32, reg1」命令を使用して汎用レジスタ「reg1」にデータをセットする場合、書き込み先の reg1 (全ビット) が事前に「0」で初期化されている必要があります。reg1 内に不定値 (「x」または「z」) が含まれていると、命令の実行結果が正しく reg1 に反映されないので注意してください (制限事項)。

なお、これは Verilog シミュレーションでだけ発生するもので、実際のチップでは問題になりません。また、ほかの MOV 命令 (「MOV reg1, reg2」, 「MOV imm5, reg2」) に関しては、このような制限事項はありません。

回避策として初期化ルーチンの先頭ですべての汎用レジスタを初期化する方法があります。レジスタを使用する前には次に示すプログラムで全レジスタを初期化してください。

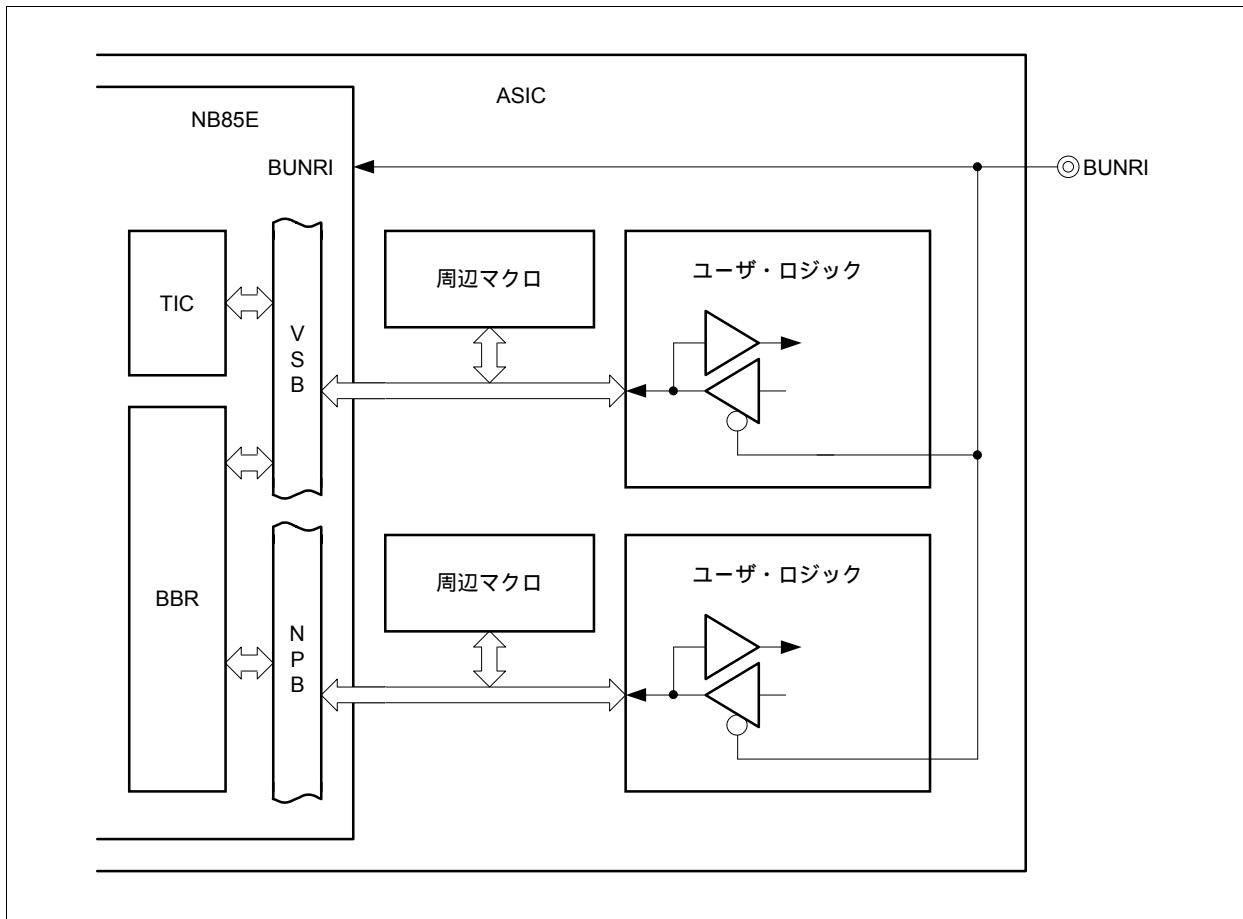
```
mov r0, r1
mov r0, r2
mov r0, r3
mov r0, r4
mov r0, r5
mov r0, r6
mov r0, r7
mov r0, r8
mov r0, r9
mov r0, r10
mov r0, r11
mov r0, r12
mov r0, r13
mov r0, r14
mov r0, r15
mov r0, r16
mov r0, r17
mov r0, r18
mov r0, r19
mov r0, r20
mov r0, r21
mov r0, r22
mov r0, r23
mov r0, r24
mov r0, r25
mov r0, r26
mov r0, r27
mov r0, r28
mov r0, r29
mov r0, r30
mov r0, r31
```

9.3 チップのBUNRIテスト時の注意

チップのBUNRIテスト時にはユーザ・ロジック（テスト機能が付いていないマクロ）が選択されないようにしてください。

ただし、テスト機能が搭載されたマクロについては、BUNRIテスト時でも選択されるようにする必要があります。詳細については、**第10章 テスト回路設計**を参照してください。

図9-1 ユーザ・ロジック設計例



9.4 タイミング調整

高速動作を行う回路の設計の際には、スキュー調整やホールド時間調整が必須になります。



スキューやホールド時間に問題がある場合、シミュレーション上では次のような症状が発生します。

特に、クロック・スキューとデータ・ラインのホールド・タイミングには注意が必要です。

- リセット解除後、命令フェッチするまでは正常であるが、ROM から数命令読み出したあとはすべて不定となる。
- 外部から読み込んだデータが不正な値を取る。
- まったくシミュレーションできない（すべての出力端子が不定）。
- 「Timing Violation」が出る（特に「\$hold」エラー）。

9.4.1 クロック・スキュー，データ・ラインのホールド違反の調整方法

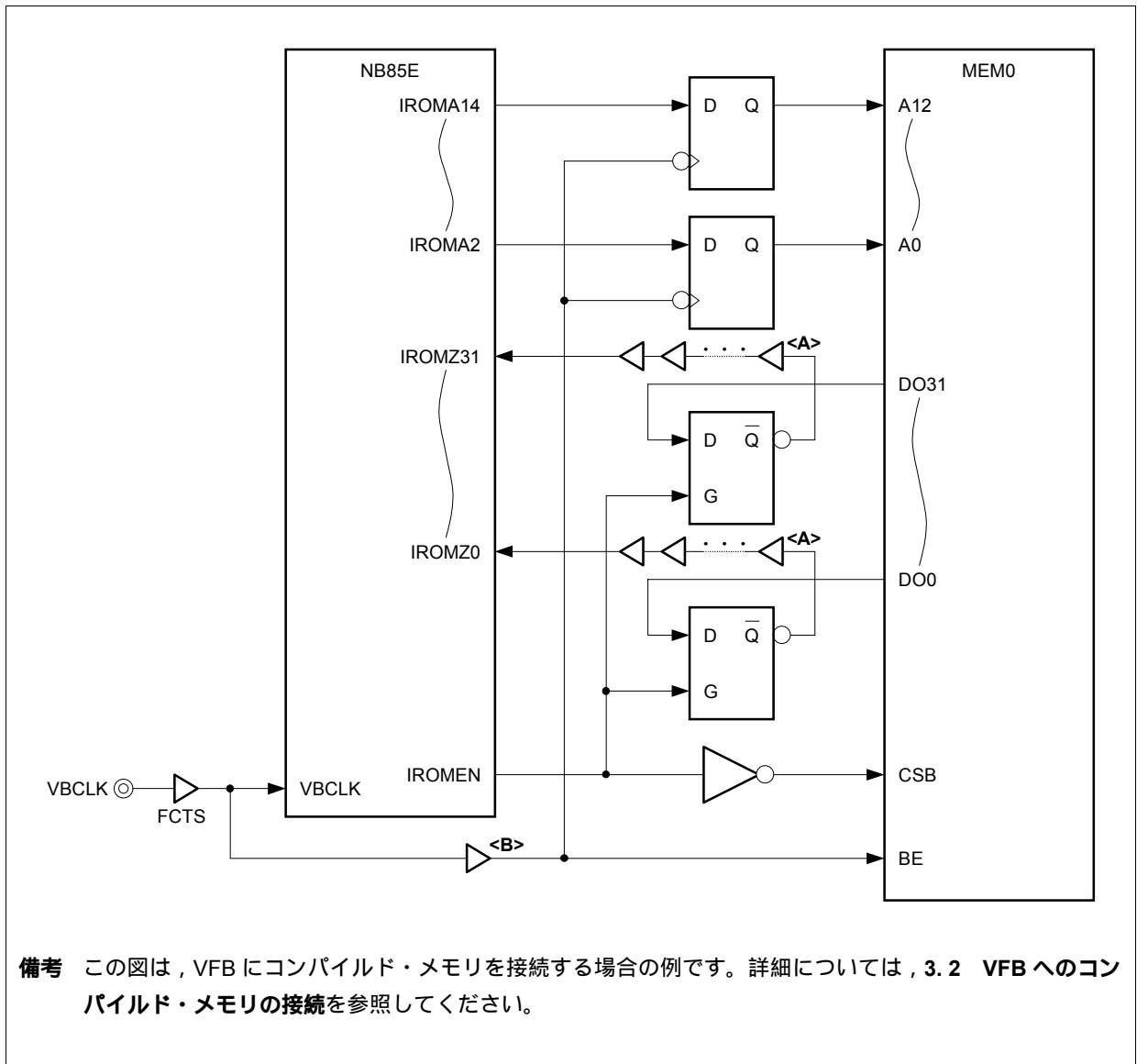
クロック・スキュー，データ・ラインのホールド時間確保は，次の2つの方法を併用して行ってください。

- データ・バスにバッファを挿入する方法（ 9 - 2 の<A>参照）
- クロック・ラインにバッファを挿入する方法（ 9 - 2 の参照）

備考 1. HDL で回路にクロック・スキュー調整用バッファを挿入する場合は，構造記述で指定してください。

2. 論理合成時の注意点については，**NEC システム LSI 設計 OPENCAD V5.4 ユーザーズ・マニュアル Design Compiler インタフェース編 (A15058J)** を参照してください。

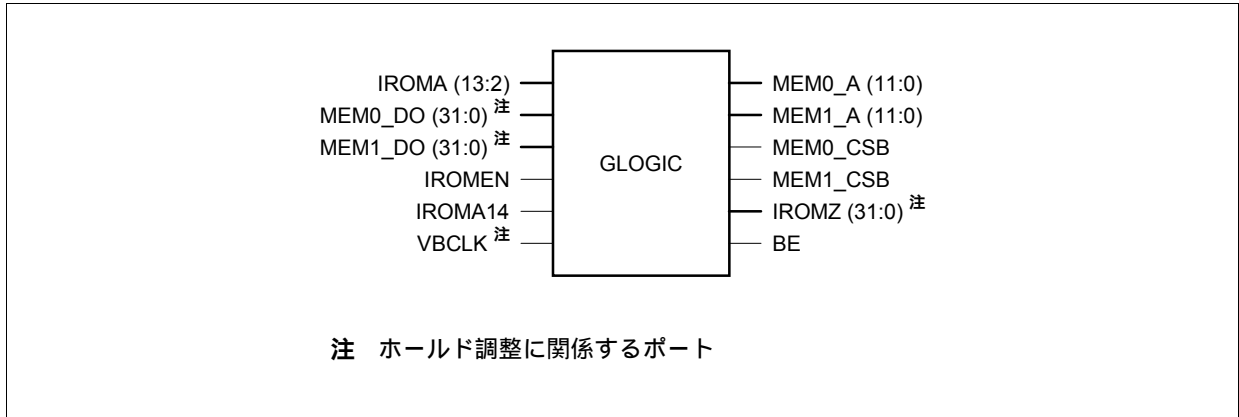
図9-2 スキュー, ホールド違反対策実施例



9.4.2 デザイン・コンパイラを使用してホールド違反を解決する方法

デザイン・コンパイラを使用して、ホールド違反を調整する場合、HDL による構造記述で回路に直接挿入する方法と論理合成スクリプトに記述する方法があります。ここでは、論理合成スクリプトで対応する場合について例を示します。

図9-2の回路からCPUとコンパイルド・メモリをつなぐ部分を切り出したモジュール・ブロックは次のようになります(モジュール名:GLOGIC)。



たとえば、CPU クロック周波数が 33MHz (周期 30ns) で、VBCLK に対する IROMZ のホールド時間が 4ns 以上必要な場合、次のようなスクリプトになります(別途、input_delay も定義する必要があります)。この例では、IROMZ に 5ns-7ns の遅延が挿入されます。

```
/* Set the current_design */
current_design GLOGIC

create_clock -period 30 -waveform {0 15} find (port, "VBCLK")
set_fix_hold find (clock, "VBCLK")
set_output_delay 23 -max -clock "VBCLK" find (port, "IROMZ")
set_output_delay -5 -min -clock "VBCLK" find (port, "IROMZ")
```

9.4.3 クロック・スキューの調整

BE 端子に入力されるクロックは、CPU に入力するクロック信号 (VBCLK) を基準にすると、クロック・スキューが存在します。この項では、VFB にコンパイルド ROM を接続する場合を例として、クロック・スキュー調整の計算方法について説明します。

(1) クロック・スキュー (VBCLK 信号に対する BE 信号のスキュー) の計算方法

CPU のクロック周波数 (f_{VBCLK})、データ入力セットアップ時間 (t_{SIDK})、ホールド時間 (t_{HKID}) とコンパイルド ROM のサイクル時間 (t_{RC})、アクセス時間 (t_{ACC})、出力ホールド時間 (t_{OH}) の関係から、VBCLK 信号に対して BE 信号のスキューがどの程度許容されるかを、次の計算で求めることができます。

$$\Delta t_{sk}^- < t_{OH} - t_{HKID} \quad : \text{データ・ホールド条件}$$

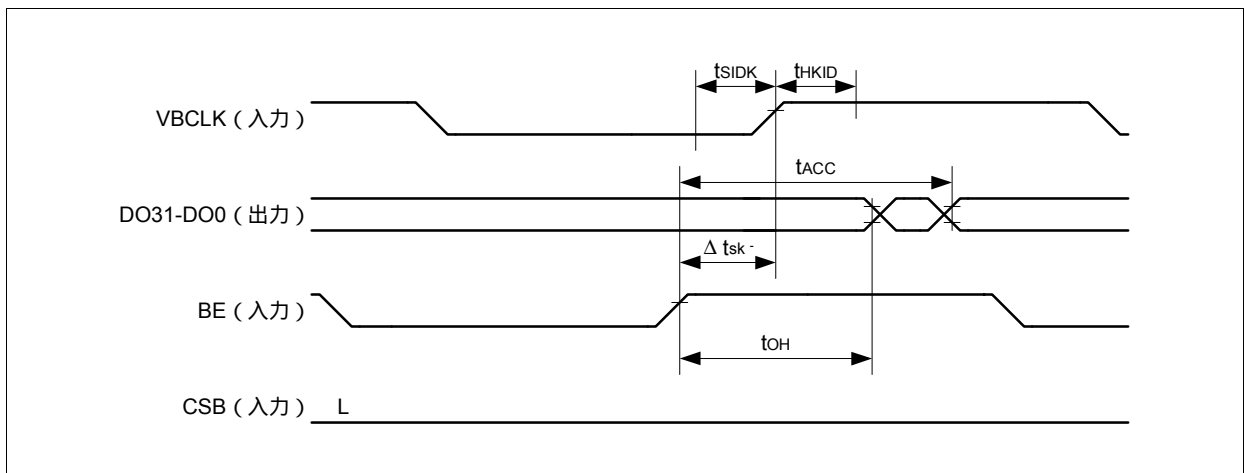
$$\Delta t_{sk}^+ < (1 / f_{VBCLK} - t_{ACC}) - t_{SIDK} \quad : \text{データ・セットアップ条件}$$

(ただし、 $t_{RC} < 1 / f_{VBCLK}$)

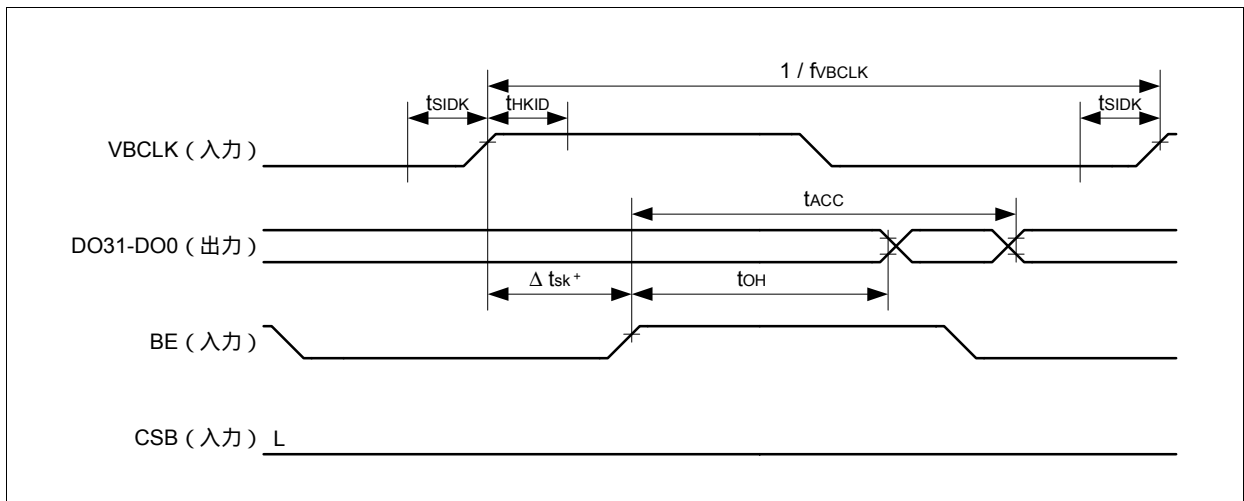
各略号の意味は、次のとおりです。

- t_{sk}^- : VBCLK に対して BE のスキューが進んでいる場合の値
- t_{sk}^+ : VBCLK に対して BE のスキューが遅れている場合の値
- t_{OH} : ROM の出力ホールド時間
- t_{RC} : ROM のサイクル時間
- t_{ACC} : ROM のアクセス時間
- t_{SIDK} : CPU データ入力セットアップ時間
- t_{HKID} : CPU データ入力ホールド時間
- f_{VBCLK} : CPU クロック周波数 (Hz)

(a) データ・ホールド条件 ($\Delta t_{sk}^- < t_{OH} - t_{HKID}$: ただし, $t_{RC} < 1 / f_{VBCLK}$)



(b) データ・セットアップ条件 ($\Delta t_{sk}^+ < (1 / f_{VBCLK} - t_{ACC}) - t_{SIDK}$: ただし, $t_{RC} < 1 / f_{VBCLK}$)



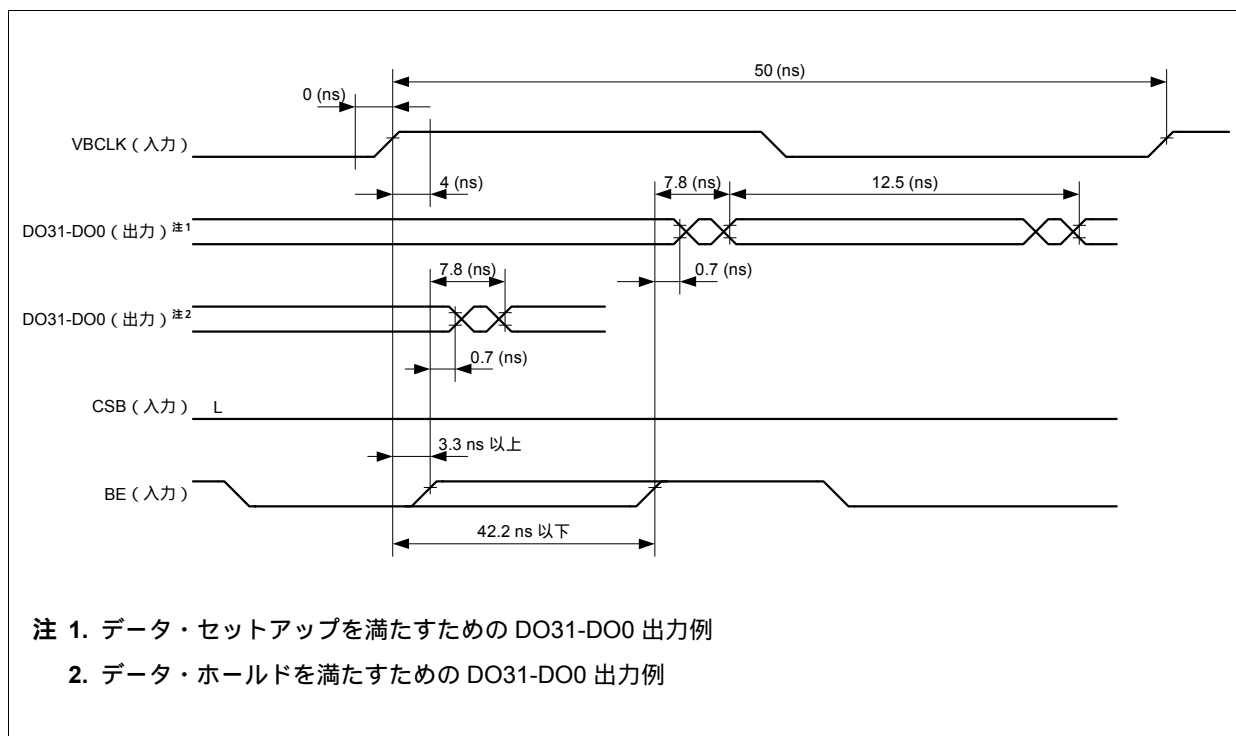
(2) クロック・スキュー許容値 (VBCLK 信号に対する BE 信号のスキュー) の算出例

たとえば, CB-9 ファミリ VX タイプ ($V_{DD} = 3.3V$), 32 ビット × 4K ワードの高速同期 ROM の場合, $t_{OH} = 0.7ns$, $t_{ACC} = 7.8ns$, $t_{RC} = 12.5ns$, VBCLK 周波数 (f_{VBCLK}) = 20MHz (周期: 50ns), $t_{SIDK} = 0ns$, $t_{HKID} = 4ns$ で配線遅延が無視できるくらい小さいとすれば, 計算式は次のようになります。

$$\Delta t_{sk}^- < t_{OH} - t_{HKID} = 0.7 - 4 = -3.3 (ns)$$

$$\Delta t_{sk}^+ < (1 / f_{VBCLK} - t_{ACC}) - t_{SIDK} = (50 - 7.8) - 0 = 42.2 (ns)$$

(ただし, $t_{RC} < 50 (ns)$)



注 1. データ・セットアップを満たすための DO31-DO0 出力例

2. データ・ホールドを満たすための DO31-DO0 出力例

この結果から, BE 信号を (VBCLK に対して) 4ns 以上遅らせる必要があることが分かります。

このようにスキューのタイミングが合わなくなる場合, 次のような対策が考えられます。

- メモリのデータ出力に遅延調整用のパッファを挿入し, 出力ホールド時間を延ばす (図 9 - 2 の<A>参照)。
- NB85E への VBCLK 入力タイミングに対してメモリ側への BE 信号入力を遅らせるために遅延調整用のパッファを挿入する (図 9 - 2 の参照)。

9.5 デバイス・ファイル

C コンパイラ/アセンブラ (CA850) , 統合ディバッガ (ID850) , システム・シミュレータ (SM850) を使用する場合は, デバイス・ファイルが必要になります。

このデバイス・ファイルは NEC 側で用意しますが, 開発環境や開発品種により内容が異なるため, 次の項目については事前に NEC までご連絡ください。

(1) 開発環境

使用する開発ツール (コンパイラ, ディバッガ, インサーキット・エミュレータ) のメーカーとツールの名称

(2) 製品仕様

- (a) CPU コア (NB85E, NB85E + NB85E901, NB85ET)
- (b) メモリ・コントローラ (NB85E500, NU85E500, NU85E502)
- (c) VDB に接続する RAM のサイズ
- (d) VFB に接続する ROM のサイズ
- (e) IFIROME, IFIROB2, IFIRA64, IFIRA32, IFIRA16, IFIMAEN, IFID256, IFINSZ1, IFINSZ0, IFIWRTH, IFIUNCH1, IFIUNCH0 端子の設定
- (f) ユーザ・ロジックの各レジスタ名と割り付けたアドレス

第 10 章 テスト回路設計

NB85E は、テスト・インタフェース・コントロール・ユニット (TIC) を内蔵しているため、テスト・バス (TBI39-TBI0, TBO34-TBO0) を介して、NB85E 自身や、周辺マクロ (命令キャッシュ、データ・キャッシュ、メモリ・コントローラ (MEMC) など) のテストを行います。

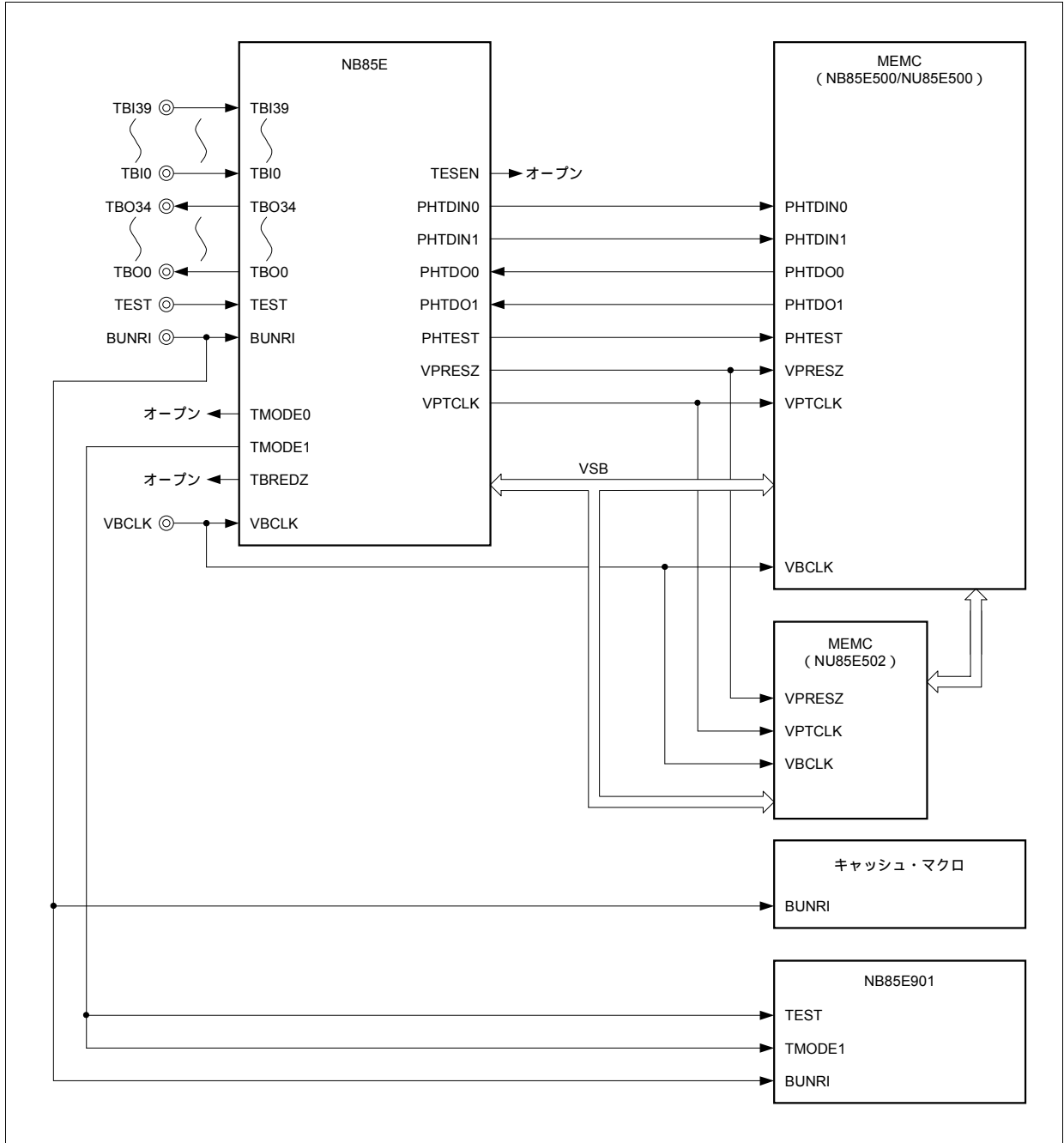
テスト・バスは、TEST 信号、BUNRI 信号がアクティブのときに有効となります。

なお、VFB、VDB に接続されるコンパイルド・メモリについては、それ自身が持つテスト・バスを用いてテストを行います。

10.1 テスト・モード時の周辺マクロ接続例

図 10 - 1 に NB85E と周辺マクロの接続例を示します。

図10 - 1 NB85Eと周辺マクロの接続例



10.2 テスト・モード時の各端子の処理

10.2.1 NB85E, NB85ET の各端子の処理

(1) テスト・モード用端子以外

(a) 入出力端子

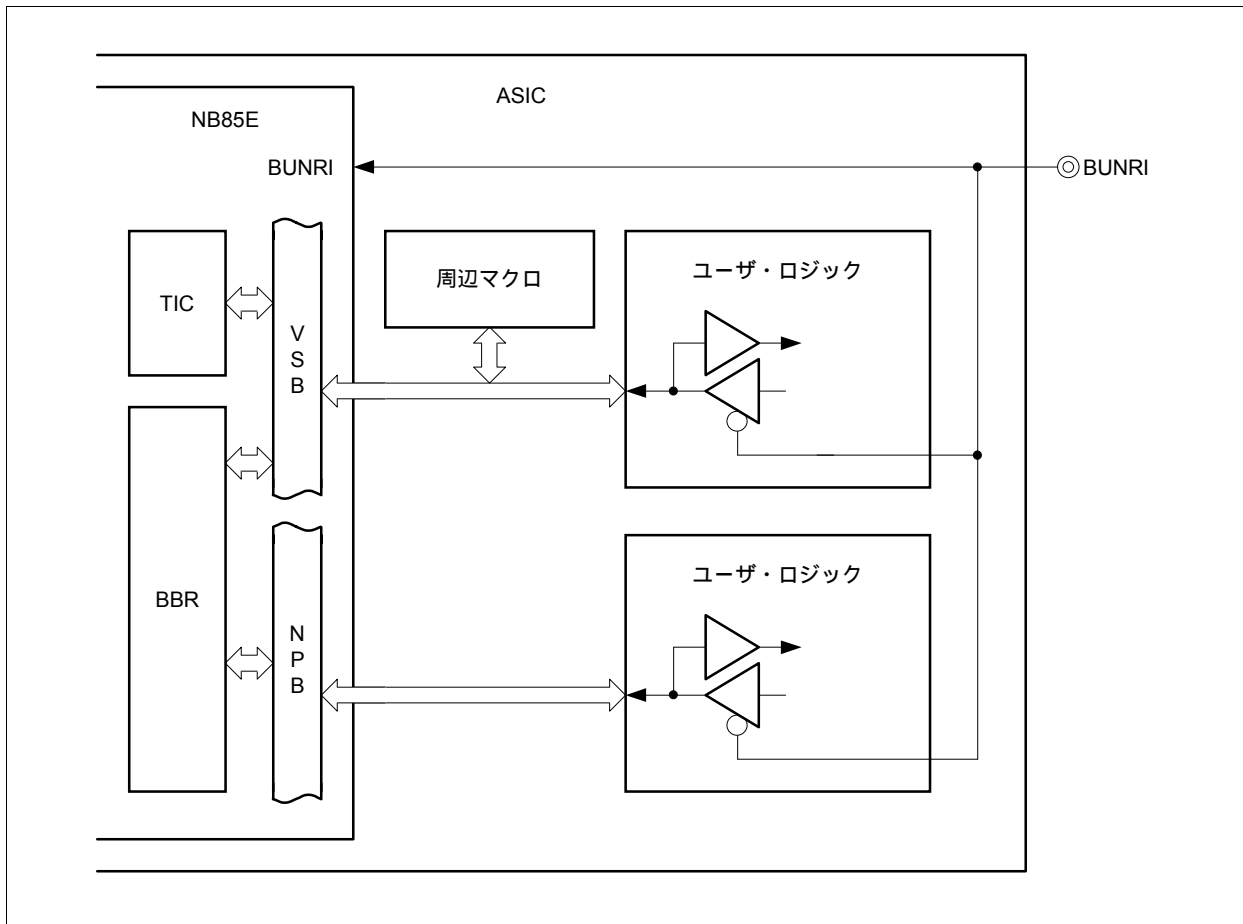
テスト・インタフェース端子経由で周辺マクロのテストを行う際に、VSB に周辺マクロとユーザ・ロジックがともに接続されていると、テスト・モード時に信号の衝突が発生する可能性があります。信号の衝突を避けるためには、周辺マクロの信号だけを有効にする必要があります。そのため、ユーザ・ロジックに接続する次の入出力端子については、テスト・モード時にハイ・インピーダンスになるような設計をユーザ・ロジック側で行ってください(図 10-2 参照)。なお、あらかじめテスト・モード時に各信号が衝突しないような設計がされている場合は、このかぎりではありません。

- | | | |
|-------------------|--------------------|--------------------|
| • VBWAIT | • VBLAST | • VBAHLD |
| • VBSEQ2-VBSEQ0 | • VBBSTR | • VDCSZ7-VDCSZ0 |
| • VDSELPZ | • VBTTYP1, VBTTYP0 | • VBSIZE1, VBSIZE0 |
| • VBBENZ3-VBBENZ0 | • VBWRITE | • VBLOCK |
| • VBCTYP2-VBCTYP0 | • VBA27-VBA0 | • VBSTZ |
| • VPD15-VPD0 | • VBD31-VBD0 | |

上記以外の入出力端子は、ノーマル・モードと同様にしてください(未使用の場合は、オープンにしてください)。

注意 NEC ではテスト・パス自動結線ツールをサポートしていますが、NB85E には対応していません。ユーザ自身でテスト・パスの結線を行ってください。

図10 - 2 ユーザ・ロジック設計例



(b) 入力端子の処理

VAREQ 端子にはロウ・レベルを入力してください。VAREQ 以外の端子は、特に処理をする必要はありません（ノーマル・モードと同様にしてください）。

(c) 出力端子の処理

特に処理をする必要はありません（ノーマル・モードと同様にしてください）。

(2) テスト・モード用端子

テスト・モード用端子は、次の内容で端子処理をしてください。

端子名	入出力	端子処理		
		MEMC が接続されている場合	キャッシュが接続されている場合	MEMC, キャッシュが接続されていない場合
PHTDOn	入力	NB85E500/NU85E500 の PHTDOn 端子に接続してください。	-	ロウ・レベルを入力してください。
PHTDInn	出力	NB85E500/NU85E500 の PHTDInn 端子に接続してください。	-	オープンにしてください。
VPRESZ	出力	NB85E500/NU85E500, NU85E502 の VPRESZ 端子に接続してください。	VPRESZ 端子に接続してください。	
VPTCLK	出力	NB85E500/NU85E500, NU85E502 の VPTCLK 端子に接続してください。	VPTCLK 端子に接続してください。	
TESEN	出力	-	-	
PHTEST	出力	NB85E500/NU85E500 の PHTEST 端子に接続してください。	-	
TMODEn, TBREDZ	出力	オープンにしてください。		

備考 n = 1, 0

(3) NB85E901 (RCU) 使用時の注意事項

NB85E に NB85E901 (RCU) を接続している場合、次に示す端子は、単体テスト・モード時に使用します。これらの端子はすべて外部端子として、チップの外に出してください。

- TBI39-TBI0^{注1}
- TBO34-TBO0^{注1}
- TEST^{注1}
- BUNRI
- DCK^{注2}
- DRSTZ^{注2}
- DMS^{注2}
- DDI^{注2}
- DDO^{注2}
- DBINT^{注1,2}

注1. ノーマル・モードで使用する端子と兼用可能です。

2. NB85E901 の端子です。

(4) NB85ET 使用時の注意事項

NB85ET の単体テスト・モードでは、テスト・バス以外に 13 本の N-Wire 型インサーキット・エミュレータ接続用端子 (DCK, DRSTZ, DMS, DDI, DDO, DBINT, EVTTRG, TRCCLK, TRCDATA3-TRCDATA0, TRCEND) も使用します。これらの端子はすべてテスト・モード時に外部端子として、チップの外に出してください。また、これらの端子は兼用端子にはしないでください (ただし、EVTTRG 端子と DBINT 端子はテスト・バス (TBI39-TBI0, TBO34-TBO0) 以外の端子と兼用可能です)。

10.2.2 MEMC の各端子の処理

(1) NB85E500

(a) 外部メモリ接続用端子

テスト・モード時も、ノーマル・モードと同様の動作をします。

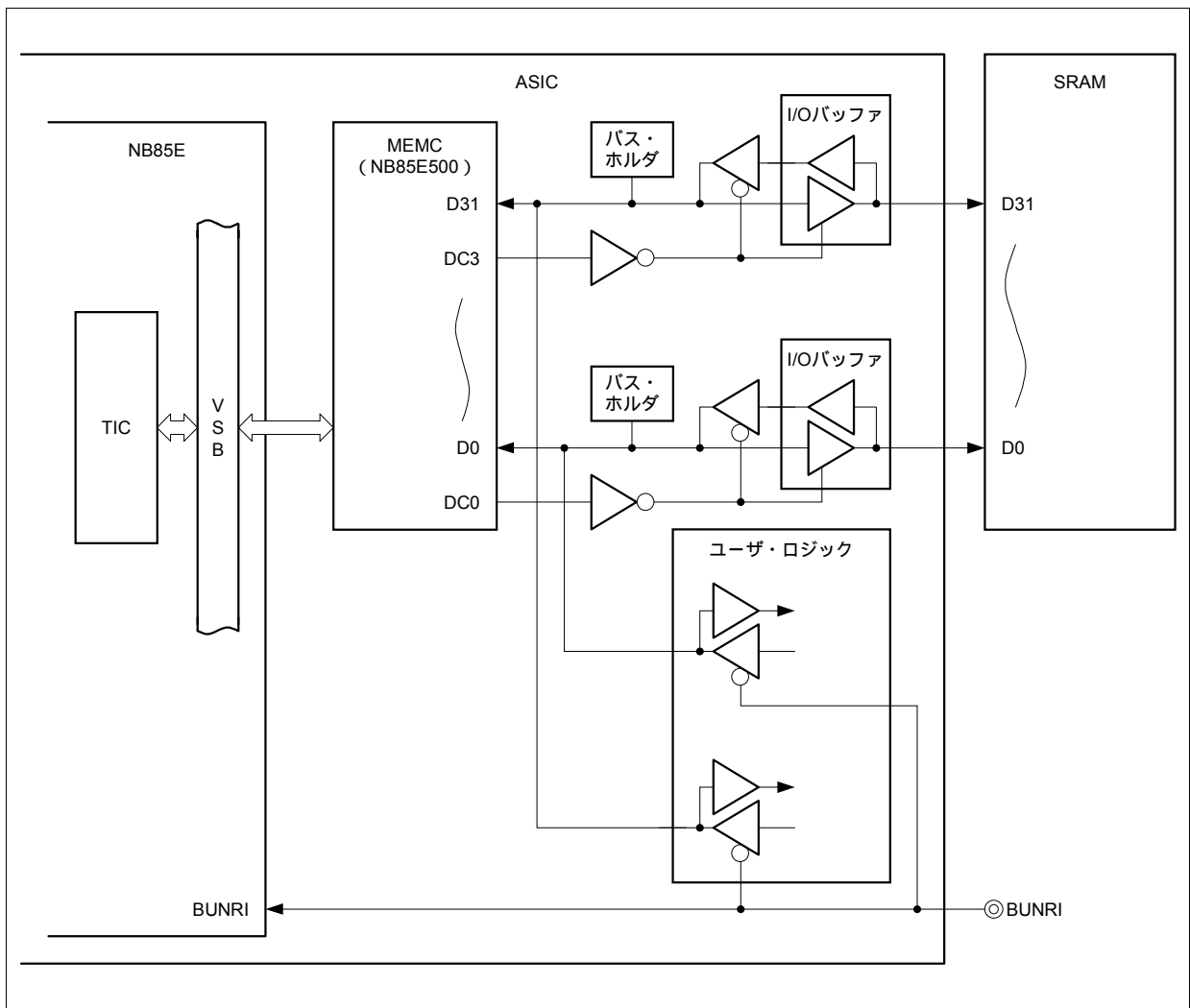
このため、データ・バス (D31-D0) にユーザ・ロジックや SRAM などが接続されている場合は、テスト・モード時にデータ・バスの信号が衝突する可能性があります。これを避けるために、テスト・モード時に D31-D0 端子がハイ・インピーダンスになるような設計をユーザ・ロジックで行ってください (図 10 - 3 参照)。

D31-D0 以外の端子はノーマル・モードと同様にしてください (未使用の場合は、メモリ・コントローラ ユーザーズ・マニュアル NB85E, NB85ET 編 (A14206J) の未使用端子の処理の項目に示されている内容で処理してください)。

なお、入力端子 (HLDRQZ, WAITZ, SELFREF) に入力される信号は無視されます。

注意 NEC ではテスト・バス自動結線ツールをサポートしていますが、NB85E には対応していません。ユーザ自身でテスト・バスの結線を行ってください。

図10 - 3 ユーザ・ロジック設計例



(b) テスト・モード用端子

図 10 - 1 で示すように NB85E に接続してください。

(c) その他の端子

ノーマル・モードと同様にしてください (未使用の場合は、**メモリ・コントローラ ユーザーズ・マニュアル NB85E, NB85ET 編 (A14206J)** の未使用端子の処理の項目に示されている内容で処理してください)。

(2) NU85E502

(a) 外部メモリ接続用端子

テスト・モード時も、ノーマル・モードと同様の動作をします。

なお、入力端子 (D31-D0) は入力される値にかかわらず無視されます。

(b) テスト・モード用端子

図 10 - 1 で示すように NB85E に接続してください。

(c) その他の端子

ノーマル・モードと同様にしてください。

10.3 テスト・バス自動結線

NB85E および NB85E 周辺マクロは、テスト・バス自動結線ツールに対応していないため、ユーザ自身によるテスト・バスの結線が必要でしたが、**10.3.1 テスト・バス自動結線方法**に示す方法により自動結線ツールを適用することができます（NB85E および NB85ET のみ自動結線ツールにより自動結線が可能です）。

NB85E 周辺マクロ（メモリ・コントローラ、命令/データ・キャッシュなど）およびユーザ・ロジックは、自動結線前の接続、または自動結線後のネット・リスト修正による対応が必要です。

また、テスト・バスの自動結線をする/しないにかかわらず、テスト・バス PINF ファイルの作成/修正が必要になります（詳細は **10.4 テスト・バス PINF ファイルの作成/修正方法**を参照してください）。

10.3.1 テスト・バス自動結線方法

(1) NB85E

通常の方法で自動結線してください。

(2) NB85ET

通常の方法で自動結線してください。ただし、N-Wire 型インサーキット・エミュレータ接続用端子（DCK, DRSTZ, DMS, DDI, DDO, DBINT, EVTTRG, TRCCLK, TRCDATA3-TRCDATA0, TRCEND）は、テスト端子と兼用しないように自動結線時に指定してください（ただし、DBINT, EVTTRG 端子はテスト・バス以外の端子と兼用可能です）。

(3) メモリ・コントローラ（NB85E500, NU85E500, NU85E502）

テスト・モード用端子がないため自動結線対象外です。

NB85E との接続をあらかじめ行ってください。NB85E500, NU85E500, NU85E502 の単体テストは、NB85E のテスト・モード用端子を使用して行います。

(4) 命令キャッシュ（NB85E212, NB85E213）、データ・キャッシュ（NB85E252, NB85E263）

BUNRI 端子はありますが、TBix, TBox 端子がないため自動結線対象外です。

BUNRI 端子をあらかじめ接続すると自動結線ができなくなるため、NB85E の TMODE1 端子を NB85E2xx の BUNRI 端子に接続してください（自動結線時は NB85E の TMODE1 信号で BUNRI 信号を代替させます）。NB85E2xx の単体テストは、NB85E のテスト・モード用端子を使用して行います。

(5) NB85E901

BUNRI, TEST 端子はありますが、TBix, TBox 端子がないため自動結線対象外です。

BUNRI, TEST 端子をあらかじめ接続すると自動結線ができなくなるため、BUNRI 端子をロウ・レベルにクランプしておき、TEST 端子を NB85E の TMODE1 端子に接続してテスト・バスの自動結線を行ってください。実行後に出力される PWC フォーマットのネット・リストを verilog ネット・リストに変換して、NB85E901 の BUNRI 端子に NB85E の BUNRI 端子に入力されている信号を接続してください。その後、再度 verilog ネット・リストを PWC フォーマットに変換してください。

NB85E901 は、NB85E のテスト・モード用端子と NB85E901 の N-wire 型インサーキット・エミュレータ接続用端子（DCK, DRSTZ, DMS, DDI, DDO, DBINT）を単体テスト時に使用します。N-Wire 型 インサーキット・エミュレータ接続用端子は外部端子として必ずチップの外に出してください。また、テスト端

子と兼用しないよう自動結線時に指定してください (DBINT 端子はノーマル・モードで使用する端子と兼用可能です)。

(6) ユーザ・ロジック

図 10 - 2, 10 - 3 で示したユーザ・ロジック設計例では、あらかじめ外部 BUNRI 端子を設定し、その BUNRI 信号を使用してユーザ・ロジックを設計しています。しかし、BUNRI 端子をあらかじめ接続してしまうと自動結線ができなくなります。したがって、NB85E901 と同様にユーザ・ロジックへの BUNRI 入力信号をロウ・レベルにクランプしておき、自動結線後のネット・リストを修正して BUNRI 信号を接続し直してください。

10.3.2 ダミー・モデルによるテスト・バス接続確認

NB85E および NB85ET はテスト・バスの自動結線後、ダミー・モデルによるシミュレーションによりテスト・バスの接続確認ができます。

ただし、このダミー・モデルによるテスト・バスの接続確認は、NB85E および NB85ET のテスト・バスが正常に接続されているかの確認だけです。

CPU コアと NB85E 周辺マクロとの接続や N-Wire 型インサーキット・エミュレータ接続用端子など通常端子側の接続確認はできません。また、NB85E および NB85ET のテスト・バスを使用した NB85E 周辺マクロの単体テストが正常に行われるかの確認もできません。したがって、NB85E コアと NB85E 周辺マクロとの接続や、VSB やメモリ・コントローラにユーザ回路が接続されている場合は、必要に応じてフルファンクション・モデルによるトータル・チップ・シミュレーションを行ってください (第 11 章 トータル・チップ・シミュレーション参照)。

10.4 テスト・バス PINF ファイルの作成 / 修正方法

NB85E の周辺マクロであるメモリ・コントローラ，命令 / データ・キャッシュ，NB85E901 は自動結線対象外のため，自動結線時に作成される PINF ファイルにそのマクロの情報が反映されません。また，NB85E901 や NB85ET の分離テストで使用する N-Wire 型 インサーキット・エミュレータ接続用端子の情報も反映されません。

この節では自動作成された PINF ファイルの修正方法を示します。また，この PINF ファイルは自動結線を行わなくても，各メガマクロの単体テストを行うためのテスト・パターンを生成するために必要です。

次に PINF ファイルの作成例を示します。

10.4.1 PINF ファイル作成例

(1) NB85E + NU85E500 + NU85E502 + NB85E901 + NB85E212 の場合

BUNRI 外部端子名 : BUNRI

テスト・モード設定外部端子名 : TMC1, TMC2

TMC1 = TMC2 = 0 (NB85E および NB85E 周辺マクロの単体テスト)

- 備考**
1. NB85E のテスト・バスを使わないメガマクロの単体テスト・モードの場合は，N-Wire 型 インサーキット・エミュレータ接続用端子の設定は必要ありません。
 2. NB85E901 の N-Wire 型 インサーキット・エミュレータ接続用端子は，そのままの端子名で外部端子になっています。
 3. *DECODER 部：マクロのテスト時に必要な端子の設定を記述しています。
 4. *TESTBUS 部：テスト信号の入出力端子を記述しています。

図10 - 4 PINFファイル作成例 (NB85E + NU85E500 + NU85E502 + NB85E901 + NB85E212) (1/2)

MACRO NB85E (MACRO1)	} NB85E 単体テスト時の端子指定
*DECODER	
TMC1 : 0	} NB85E の単体テスト・モードを指定
TMC2 : 0	
BUNRI : 1	} NB85E901 の N-Wire 端子をインアクティブ・レベルに設定 (ただし、リセット端子はアクティブ) (DBINT 端子は専用端子にしない場合でも、マクロ分離テスト時に必要なため、兼用端子として設定してください)
DCK : 1	
DMS : 1	
DDI : 1	
DRSTZ : 0	
DBINT : 0	
*TESTBUS	
xxxx : TBI0	} テスト・バス端子を記述 (xxxx : 各兼用外部端子名)
: :	
xxxx : TBI39	
xxxx : TBO0	
: :	
xxxx : TBO34	
*END	
MACRO NU85E500 (MACRO2)	} NU85E500 単体テスト時の端子指定
*DECODER	
TMC1 : 0	} NB85E の単体テスト・モードを指定
TMC2 : 0	
BUNRI : 1	} NB85E901 の N-Wire 端子をインアクティブ・レベルに設定 (ただし、リセット端子はアクティブ) (DBINT 端子は専用端子にしない場合でも、マクロ分離テスト時に必要なため、兼用端子として設定してください)
DCK : 1	
DMS : 1	
DDI : 1	
DRSTZ : 0	
DBINT : 0	
*TESTBUS	
xxxx : TBI0	} テスト・バス端子を記述 (xxxx : 各兼用外部端子名)
: :	
xxxx : TBI39	
xxxx : TBO0	
: :	
xxxx : TBO34	
*END	
MACRO NU85E502Cn (MACRO3)	} NU85E502 単体テスト時の端子指定 (n = 7-0 (n は NU85E500 が対応するチップ・セレクト領域を設定してください))
*DECODER	
TMC1 : 0	} NB85E の単体テスト・モードを指定
TMC2 : 0	
BUNRI : 1	} NB85E901 の N-Wire 端子をインアクティブ・レベルに設定 (ただし、リセット端子はアクティブ) (DBINT 端子は専用端子にしない場合でも、マクロ分離テスト時に必要なため、兼用端子として設定してください)
DCK : 1	
DMS : 1	
DDI : 1	
DRSTZ : 0	
DBINT : 0	
*TESTBUS	
xxxx : TBI0	} テスト・バス端子を記述 (xxxx : 各兼用外部端子名)
: :	
xxxx : TBI39	
xxxx : TBO0	
: :	
xxxx : TBO34	
*END	
MACRO NB85E212 (MACRO4)	} NB85E212 単体テスト時の端子指定
*DECODER	
TMC1 : 0	} NB85E の単体テスト・モードを指定
TMC2 : 0	
BUNRI : 1	} NB85E901 の N-Wire 端子をインアクティブ・レベルに設定 (ただし、リセット端子はアクティブ) (DBINT 端子は専用端子にしない場合でも、マクロ分離テスト時に必要なため、兼用端子として設定してください)
DCK : 1	
DMS : 1	
DDI : 1	
DRSTZ : 0	
DBINT : 0	

図10 - 4 PINFファイル作成例 (NB85E + NU85E500 + NU85E502 + NB85E901 + NB85E212) (2/2)

```

*TESTBUS
  xxxx : TBI0
  :
  xxxx : TBI39
  xxxx : TBO0
  :
  xxxx : TBO34
} テスト・バス端子を記述 (xxxx : 各兼用外部端子名)

*END
MACRO NB85E901 ( MACRO5 ) } NB85E901 単体テスト時の端子指定
*DECODER
  TMC1 : 0
  TMC2 : 0
  BUNRI : 1
} NB85E の単体テスト・モードを指定

*TESTBUS
  xxxx : TBI0
  :
  xxxx : TBI39
  xxxx : TBO0
  :
  xxxx : TBO34
} テスト・バス端子を記述 (xxxx : 各兼用外部端子名)
  DCK : DCK
  DMS : DMS
  DDI : DDI
  DRSTZ : DRSTZ
  DBINT : DBINT
  DBO : DBO
} NB85E901 の単体テスト時は TESTBUS 部に N-Wire の端子 (入力と出力の両方) をすべて記入

*END
MACRO Nxxxxxxx ( MACRO6 ) } Nx85Exxx 以外のマクロ単体テスト時の端子指定
*DECODER
  TMC1 : 0
  TMC2 : 1
  BUNRI : 1
} Nxxxxxxx の単体テスト・モードを指定

*TESTBUS
  xxxx : TBI0
  :
  xxxx : TBI20
  xxxx : TBO0
  :
  xxxx : TBO20
} Nxxxxxxx のテスト・バス端子を記述 (xxxx : 各兼用外部端子名)

*END

```


(2) NB85ET + NU85E500 + NU85E502 + NB85E212 の場合

BUNRI 外部端子名 : BUNRI

テスト・モード設定外部端子名 : TMC1, TMC2

TMC1 = TMC2 = 0 (NB85ET および NB85ET 周辺マクロの単体テスト)

- 備考 1.** NB85E のテスト・パスを使わない分離テスト・モードの場合は、N-Wire 型 インサーキット・エミュレータ接続用端子の設定は必要ありません。
- 2.** NB85E901 の N-Wire 型 インサーキット・エミュレータ接続用端子は、そのままの端子名で外部端子になっています。
- 3.** *DECODER 部：マクロのテスト時に必要な端子の設定を記述しています。
- 4.** *TESTBUS 部：テスト信号の入出力端子を記述しています。

図10 - 5 PINFファイル作成例 (NB85ET + NU85E500 + NU85E502 + NB85E212) (1/2)

MACRO NB85ET (MACRO1)	} NB85ET 単体テスト時の端子指定	
*DECODER		
TMC1 : 0	} NB85ET の単体テスト・モードを指定	
TMC2 : 0		
BUNRI : 1		
*TESTBUS		
xxxx : TBI0	} テスト・バス端子を記述 (xxxx : 各兼用外部端子名)	
: :		
xxxx : TBI39		
xxxx : TBO0		
: :	} NB85ET の単体テスト時は TESTBUS 部に N-Wire 端子 (入力と出力の両方) をすべて記述 (DBINT, EVTTRG 端子は専用端子にしない場合でも、マクロ分離テスト時に必要なため、専用端子としない場合も兼用端子として設定してください)	
xxxx : TBO34		
DCK : DCK		
DMS : DMS		
DDI : DDI		
DDO : DDO		
DRSTZ : DRSTZ		
DBINT : DBINT		
TRCCLK : TRCCLK		
TRCDATA0 : TRCDATA0		
TRCDATA1 : TRCDATA1		
TRCDATA2 : TRCDATA2		
TRCDATA3 : TRCDATA3		
TRCEND : TRCEND		
EVTTRG : EVTTRG		
*END		
MACRO NU85E500 (MACRO2)	} NU85E500 単体テスト時の端子指定	
*DECODER		
TMC1 : 0	} NB85ET の単体テスト・モードを指定	
TMC2 : 0		
BUNRI : 1		
DCK : 1	} NB85ET の N-Wire 端子をインアクティブ・レベルに設定 (ただし、リセット端子はアクティブ)	
DMS : 1		
DDI : 1		
DRSTZ : 0		
DBINT : 0	} NB85ET の単体テスト時は TESTBUS 部に N-Wire 端子 (入力と出力の両方) をすべて記述 (DBINT, EVTTRG 端子は専用端子にしない場合でも、マクロ分離テスト時に必要なため、兼用端子として設定してください)	
*TESTBUS		
xxxx : TBI0		
: :		
xxxx : TBI39		
xxxx : TBO0		
: :		
xxxx : TBO34		
*END		
MACRO NU85E502Cn (MACRO3)		} NU85E502 単体テスト時の端子指定 (n = 7-0 (n は NU85E500 が対応するチップ・セレクト領域を設定してください))
*DECODER		
TMC1 : 0		} NB85ET の単体テスト・モードを指定
TMC2 : 0		
BUNRI : 1		
DCK : 1	} NB85ET の N-Wire 端子をインアクティブ・レベルに設定 (ただし、リセット端子はアクティブ)	
DMS : 1		
DDI : 1		
DRSTZ : 0		
DBINT : 0	} NB85ET の単体テスト時は TESTBUS 部に N-Wire 端子 (入力と出力の両方) をすべて記述 (DBINT, EVTTRG 端子は専用端子にしない場合でも、マクロ分離テスト時に必要なため、兼用端子として設定してください)	
*TESTBUS		
xxxx : TBI0		
: :		
xxxx : TBI39		
xxxx : TBO0		
: :		
xxxx : TBO34		
*END		

図10 - 5 PINFファイル作成例 (NB85ET + NU85E500 + NU85E502 + NB85E212) (2/2)

```

MACRO NB85E212 ( MACRO4 )      } NB85E212 単体テスト時の端子指定
  *DECODER
    TMC1 : 0
    TMC2 : 0
    BUNRI : 1
    DCK : 1
    DMS : 1
    DDI : 1
    DRSTZ : 0
    DBINT : 0
  } NB85ET の単体テスト・モードを指定
  } NB85ET の N-Wire 端子をインアクティブ・レベルに設定
  } (ただし、リセット端子はアクティブ)
  } (DBINT 端子は専用端子にしない場合でも、マクロ分離テスト時に必要なため、兼用端子として設定してください)
  *TESTBUS
    xxxx : TBI0
    :
    xxxx : TBI39
    xxxx : TBO0
    :
    xxxx : TBO34
  } テスト・バス端子を記述 (xxxx : 各兼用外部端子名)
  *END
MACRO Nxxxxxxx ( MACRO6 )     } Nx85Exxx 以外のマクロ単体テスト時の端子指定
  *DECODER
    TMC1 : 0
    TMC2 : 1
    BUNRI : 1
  } Nxxxxxxx の単体テスト・モードを指定
  *TESTBUS
    xxxx : TBI0
    :
    xxxx : TBI20
    xxxx : TBO0
    :
    xxxx : TBO20
  } Nxxxxxxx のテスト・バス端子を記述 (xxxx : 各兼用外部端子名)
  *END
    
```

第 11 章 トータル・チップ・シミュレーション

トータル・チップ・シミュレーションとは、NB85E を動作させて CBIC 全体で行うシミュレーションのことです。トータル・チップ・シミュレーションの目的は、次の 2 点です。

(1) 接続チェック用テスト・パターンの作成

マクロ間の接続チェック、マクロとユーザ・ロジック間の接続チェック、およびマクロ、またはユーザ・ロジックと外部端子間の接続チェックを行うテスト・パターン（テストで選別するためのテスト・パターン）を作成します。

なお、NB85E と周辺マクロ（メモリ・コントローラ、キャッシュ、NB85E901）間、およびメモリ・コントローラ（NB85E500/NU85E500 と NU85E502）間の接続チェックをすべてユーザ側で行うことは困難になります。このため、NEC で準備する各マクロごとのテスト・バスによる単体テスト・パターンをフルファンクション・モデルで Verilog シミュレーションを実行することにより、接続チェックを行うことができます。

単体テスト・パターンを要求する場合には NEC までお問い合わせください。

(2) マクロ間のタイミング検証

マクロ間、マクロとユーザ・ロジック間、およびマクロまたはユーザ・ロジックと外部端子間のタイミングをチェックします。

11.1 接続チェック用テスト・パターンの作成

接続チェック用のテスト・パターンを作成するために、プログラムを参照しながら CPU のリード・タイミングに合わせて必要な命令パターンを入力することは非常に困難です。そのため、CBIC 外部に仮想的に ROM を配置して（仮想 ROM）、そこにテスト用プログラム[※]を割り付けます。

CPU が仮想 ROM 上のテスト用プログラムを読み込んでシミュレーションを実行し、CBIC の全端子をダンプした結果を接続チェック用のテスト・パターンとして切り出します。

ここでは次の 3 つの場合について、テスト・パターンの作成方法を説明します。

- VFB に ROM が接続されていない場合
- VFB に ROM が接続され、VSB に仮想 ROM を接続してテスト・パターンを作成する場合
- VFB に ROM が接続され、その ROM にテスト用プログラムを書き込む場合

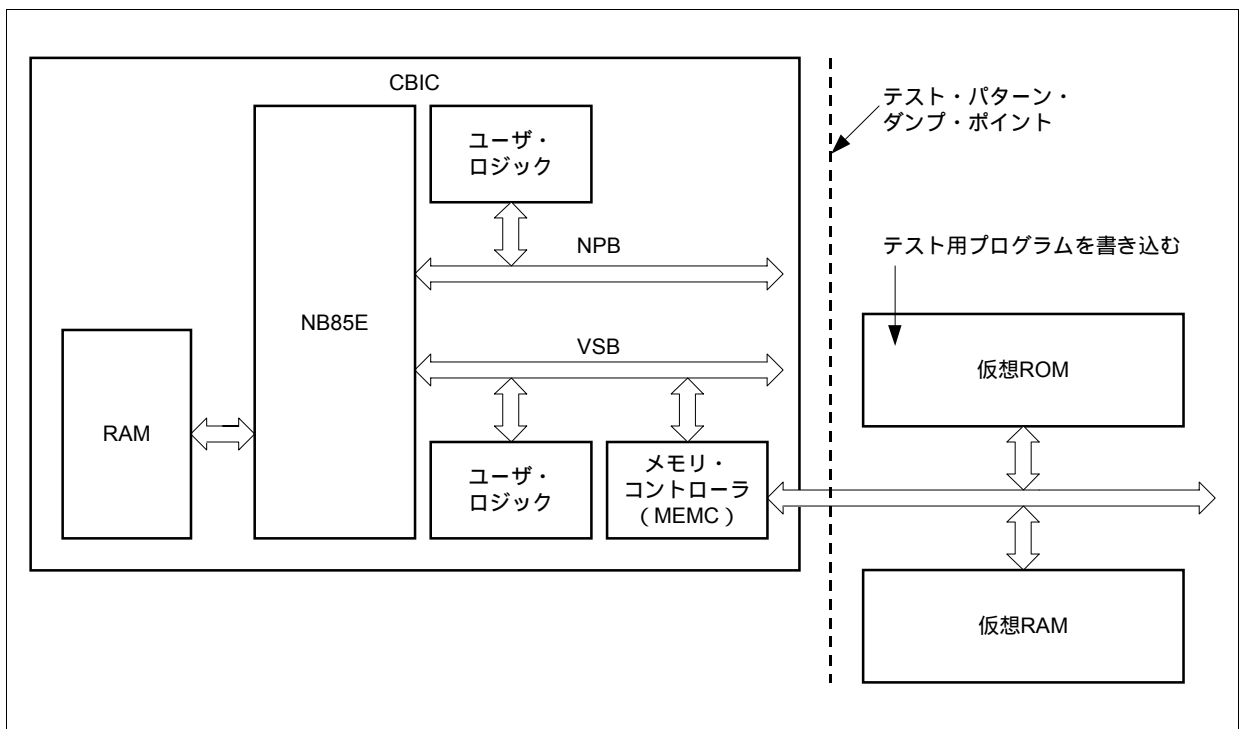
注 この章では、「テスト用プログラム」をマクロ間の接続チェック、マクロとユーザ・ロジック間の接続チェック、およびマクロ、またはユーザ・ロジックと外部端子間の接続チェックを行うためのプログラムと定義します。

(1) VFB に ROM が接続されていない場合

CBIC 外部にテスト用プログラムを書き込んだ仮想 ROM を接続します。

CBIC の全端子をダンプしてテスト・パターンとして切り出します。

図11-1 トータル・チップ・シミュレーション用ブロック図 (VFBにROMがない場合)



(2) VFB に ROM が接続されている場合

プログラム ROM を VFB に接続した場合の接続チェック用テスト・パターン作成に関して、VFB の ROM にテスト用プログラムを書き込まない場合と、書き込む場合に分けて説明します。

(a) VFB の ROM にテスト用プログラムを書き込まない場合

(仮想 ROM を接続してテスト・パターンを作成)

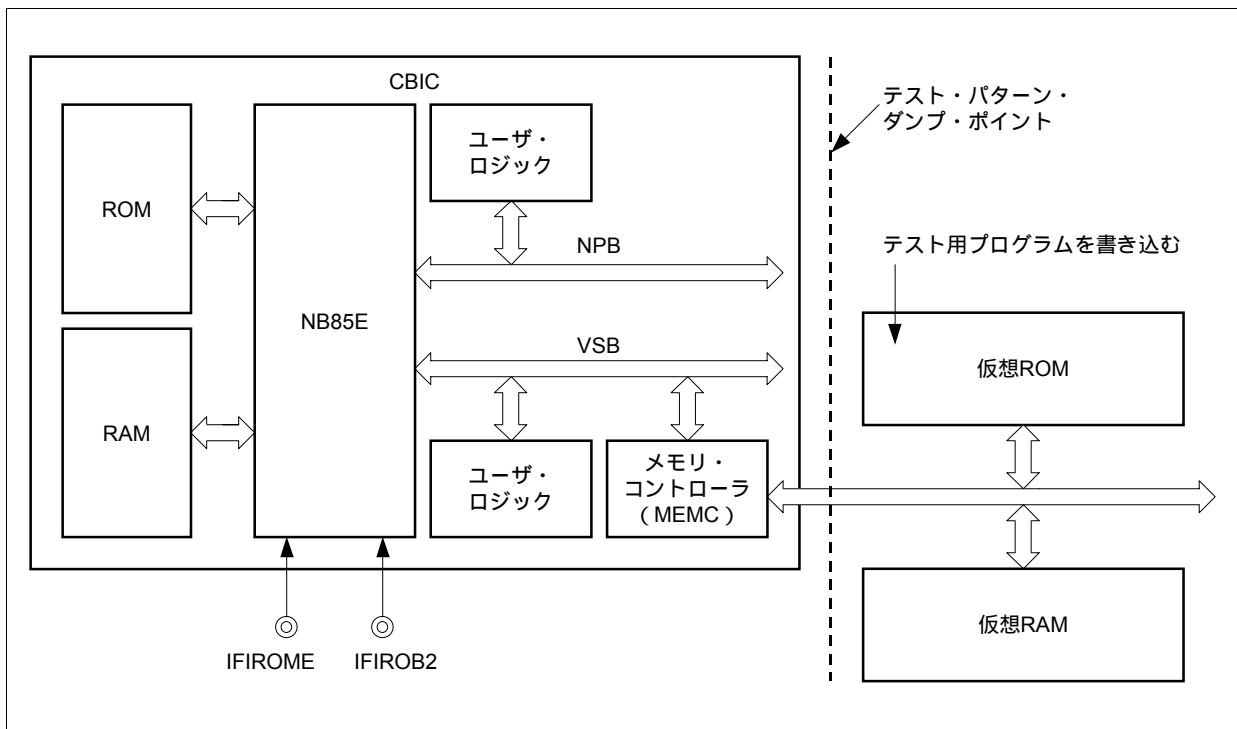
図 11 - 2 に示すように、VFB に ROM がない場合と同様に CBIC 外部に仮想 ROM を配置し、CBIC の全端子をダンプしてテスト・パターンとして切り出します。

テスト・パターンの作成は IFIROME = 0, IFIROB2 = 1 に設定し、テスト用プログラムを 100000H 番地に割り当てて行ってください。

なお、仮想 RAM は必須ではありません。

注意 出荷テストの際に NB85E を ROM レス・モードに切り替える必要があるため、CBIC の外部端子として IFIROME, IFIROB2 端子を設けてください。

図11 - 2 トータル・チップ・シミュレーション用ブロック図
(VFBにROMがあり、VSBに仮想ROMを接続してテスト・パターンを作成する場合)



(b) VFB に接続された ROM にテスト用プログラムを書き込む場合

VFB の ROM にマクロ間の接続や、マクロとユーザ・ロジックの接続検証が行えるようなテスト用プログラムを書き込み、テスト時だけ、このテスト用プログラムを実行するような設定にしておきます。

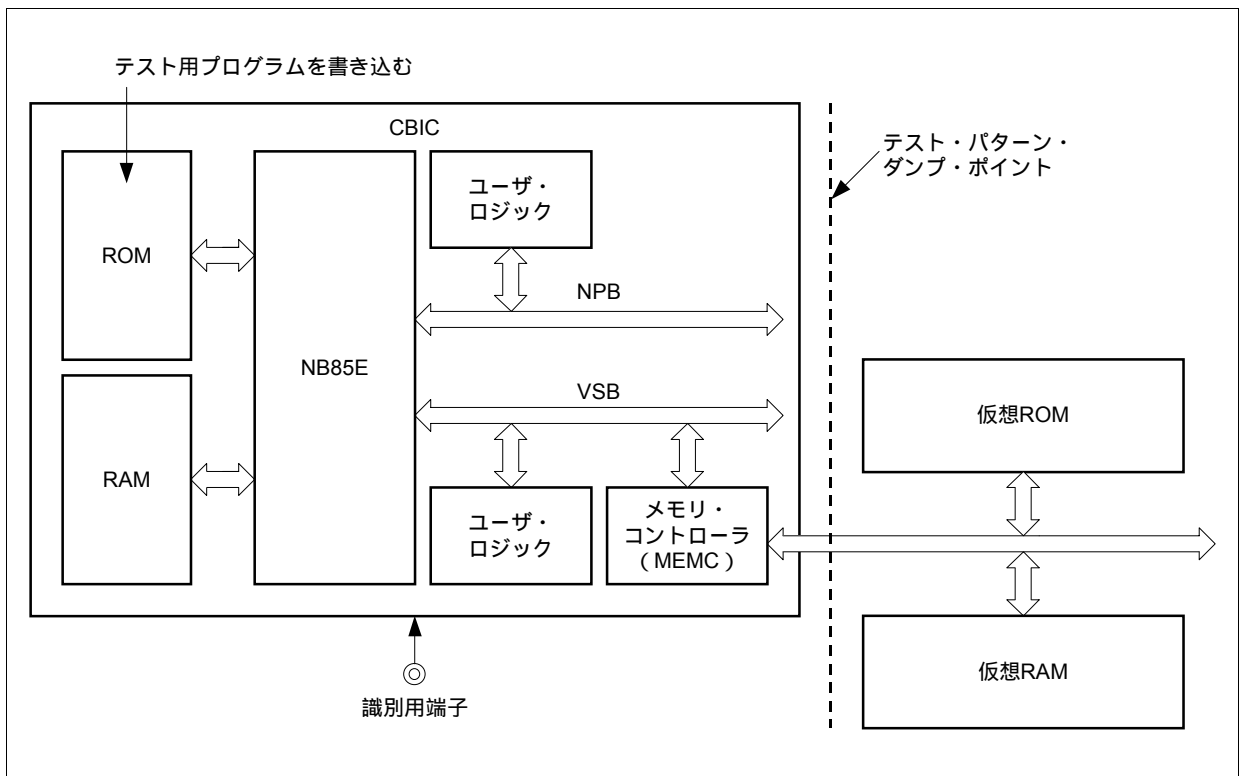
VFB の ROM にテスト用のプログラムを書き込むため、実動作プログラム^註は、その分、削減しなければなりません。

この場合の利点は主に次のとおりです。

- 外部バスを CBIC の端子としていない場合、テスト用の端子数を削減（仮想 ROM を読み出す場合、アドレス・バス、データ・バスなどを外部に出力する必要があります）
- 1 アクセス当たりのクロック数とバス幅の違いによるテスト・パターン数の減少

注 この章では、「実動作プログラム」をターゲット・システムを動作させるためのプログラムと定義します。

図11-3 トータル・チップ・シミュレーション用ブロック図
(VFBにROMがあり、そのROMにテスト用プログラムを書き込む場合)



テスト用プログラム実行と、実動作プログラム実行の識別方法として、たとえば、識別用端子を設けておいて NB85E の初期化を行ったあと、識別用端子のチェックをするようにしておきます。識別用端子のレベルに応じて、テスト用プログラム、実動作プログラムへジャンプさせ、各プログラムを実行させます。

なお、この場合の仮想 ROM、仮想 RAM は必須ではありません。

11.2 マクロ間のタイミング検証

シミュレーションでは、マクロ間接続などのファンクション確認のほかに、リアルタイムの動作検証（リアルタイム・シミュレーション）を行います。NB85E とその他のマクロ、NB85E とユーザ・ロジック、その他のマクロとユーザ・ロジックなど、それぞれの回路間でタイミング的に問題がないかを検証します。

NB85E を動作させるために、タイミング検証用のテスト・プログラムが必要となります。タイミング検証用のテスト・プログラムは CBIC 内に ROM がある場合は ROM に、ROM がない場合は、仮想 ROM に割り付けてシミュレーションを行います（タイミング検証用のテスト・プログラムは、シミュレーション時だけ使用します。実際に CBIC 内の ROM に入れる必要はありません）。

なお、リアルタイム・シミュレーションでは、MIN./MAX.シミュレーションで遅延差による不一致が発生すると予想されます。この場合、タイミング・エラーがないか、ファンクション的に間違いはないかをチェックして、不一致部分に問題がないことを確認してください。

第 12 章 ROM コードの作成

NB85E に接続する ROM はコンパイルド・タイプの ROM (コンパイルド ROM) を使用します。
コンパイルド ROM 使用時は、次の点に注意してください。

(1) サインオフ時の ROM コード・フォーマット

NEC で受注可能なフォーマットは NINCF フォーマットです。

NINCF フォーマットの詳細については、**CB-9 ファミリ VX/VM タイプ 設計マニュアル メモリ・マクロ (コンパイルド・タイプ) 編 (A12982J)** を参照してください。

(2) ROM コードを複数のコンパイルド ROM に割り付ける場合

ROM コードを複数のコンパイルド ROM に割り付ける場合は、NEC にご相談ください。

(3) ROM コードを組み込んだシミュレーション方法

詳細は **NEC システム LSI 設計 OPENCAD V5.4 ユーザーズ・マニュアル Verilog-XL インタフェース編 (A15052J)** を参照してください。

付 録 改版履歴

前版（第2版）で改訂された主な箇所を次に示します。なお、「箇所」欄に示すページは、前版でのページを示しています。

(1) 第1版 第2版

箇所	内容
全般	<ul style="list-style-type: none"> • NPB 周辺マクロの品名を「NANPxxx」→「QLNPBxxx」に変更 • MEMC の NB85E501 に関する記述を削除 • MEMC の品名を「NB85E502」→「NU85E502」に変更
p.19-22	第2章 クロック制御回路の接続 変更
p.29	図3-6 VFB へのコンパイルド ROM の接続例 修正
p.48	図5-1 NB85E, MEMC, 外部メモリ (SRAM, SDRAM) の接続例 修正
p.57,58	5.3 SDRAM との接続 データ・キャッシュ接続に関する記述を削除
p.72	図6-4 アドレス・デコーダの接続例 修正
p.75	図6-7 ユーザ・ロジックの接続例 修正
p.76	図6-9 ユーザ・ロジックの HDL 作成例 修正
p.78	図6-11 リトライ機能付きユーザ・ロジックの HDL 作成例 修正
p.92	図8-4 IE 接続推奨回路例 (NB85E + RCU (NB85E901)) 備考2を修正
p.93	図8-5 IE 接続推奨回路例 (NB85ET) 修正 備考2,3を修正
p.103	9.5 デバイス・ファイル 修正
p.109	10.2.1(3) NB85E901 (RCU) 使用時の注意事項, (4) NB85ET 使用時の注意事項 修正

[メモ]

— お問い合わせ先 —

【技術的なお問い合わせ先】

NEC半導体テクニカルホットライン
(電話：午前 9:00～12:00，午後 1:00～5:00)

電話 : 044-435-9494
FAX : 044-435-9608
E-mail : info@lsi.nec.co.jp

【営業関係お問い合わせ先】

第一販売事業部

東京 (03)3798-6106, 6107,
6108
大阪 (06)6945-3178, 3200,
3208, 3212
広島 (082)242-5504
仙台 (022)267-8740
郡山 (024)923-5591
千葉 (043)238-8116

第二販売事業部

東京 (03)3798-6110, 6111,
6112
立川 (042)526-5981, 6167
松本 (0263)35-1662
静岡 (054)254-4794
金沢 (076)232-7303
松山 (089)945-4149

第三販売事業部

東京 (03)3798-6151, 6155, 6586,
1622, 1623, 6156
水戸 (029)226-1702
前橋 (027)243-6060
鳥取 (0857)27-5313
太田 (0276)46-4014
名古屋 (052)222-2170, 2190
福岡 (092)261-2806

【資料の請求先】

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

【NECエレクトロニクス デバイス ホームページ】

NECエレクトロニクスデバイスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.ic.nec.co.jp/>

アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] CB-9 ファミリ VX/VM タイプ 設計マニュアル(暫定) NB85E, NB85ET 編
(A14335JJ3V1DM00 (第3版))

[お名前など] (さしつかえない範囲で)

御社名(学校名, その他) ()
ご住所 ()
お電話番号 ()
お仕事の内容 ()
お名前 ()

1. ご評価(各欄に をご記入ください)

項 目	大変良い	良 い	普 通	悪 い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン, 字の大きさなど					
その他()					
()					

2. わかりやすい所(第 章, 第 章, 第 章, 第 章, その他)
理由 []

3. わかりにくい所(第 章, 第 章, 第 章, 第 章, その他)
理由 []

4. ご意見, ご要望

5. このドキュメントをお届けしたのは
NEC 販売員, 特約店販売員, その他 ()

ご協力ありがとうございました。

下記あてに FAX で送信いただくか, 最寄りの販売員にコピーをお渡しください。

日本電気(株) NEC エレクトロニクス
半導体テクニカルホットライン
FAX : (044) 435-9608

2000.6