AMBA Decoder

Data Sheet

ARM®

ARM DDI 0042C

AMBA Decoder
Data Sheet

Release Information

The following changes have been made to this document.

Change History

Date	Issue	Confidentiality	Change
October 1995	A-01	Non-Confidential	Created
December 1995	A-02	Non-Confidential	Description Update
January 1996	B	Non-Confidential	Edited for style and content
April 1997	C	Non-Confidential	Minor edits

Proprietary Notice

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

Contents
AMBA Decoder Data Sheet

Chapter 1
AMBA Decoder

The following sections describe the function of the AMBA decoder.

1.1 Overview

The decoder performs three functions:

- It generates the slave select signals (DSELx) for each of the bus slaves, indicating that a transfer to that slave is required.
- It generates the slave response signals (BWAIT, BLAST and BERROR) during Address-only transfers, when no slave is selected.
- It can act as a simple protection unit which prevents attempts to access an illegal or protected area of the memory map.

Two implementations of the decoder are provided:

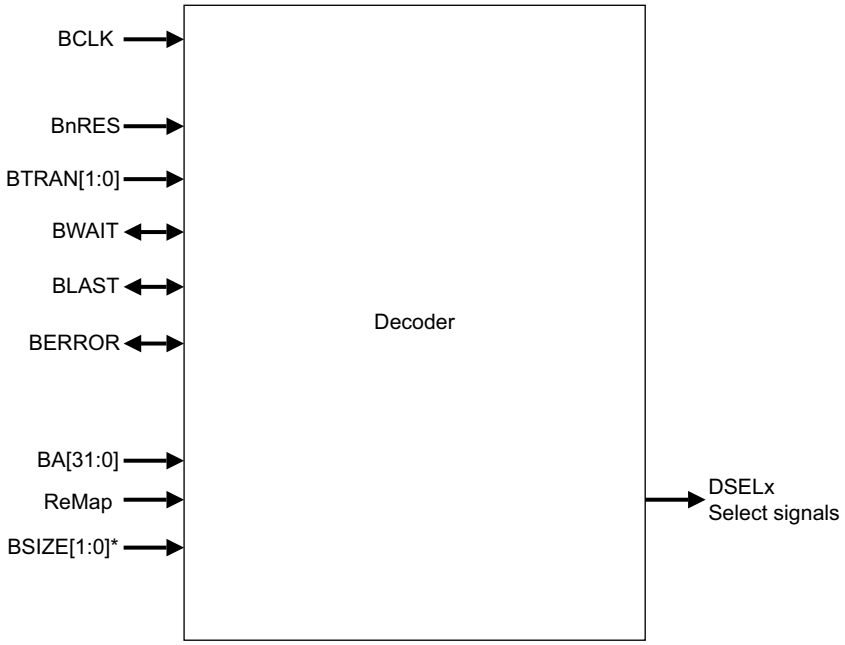
- Decoder with decode cycles. This is the default model. This implementation automatically inserts a decode cycle:
  - — at the start of a non-sequential transfer
  - — on a sequential transfer when **BLAST** has been asserted
  - — when 1KB memory boundaries are reached

This decoder is used in fast systems where the decoder might not have enough time to decode the address and assert the corresponding DSEL signal in a single clock HIGH phase.

- Decoder without decode cycles. This implementation attempts to decode the address bus on every cycle, and therefore will only be suitable for slow systems where this can be safely achieved.

Every system must have a default bus master which is granted use of the bus during reset, when no other bus master requires the bus.

A block diagram of the decoder is shown below:



* only used by the decoder with decode cycles

Figure 1-1 Decoder block diagram

1.2 Signal description

This section describes the signals that interface to the decoder.

Table 1-1 Signal descriptions for the decoder block

Name	Type	Description
BA[31:0] Address Bus	In	This is the system address bus, which is driven by the bus master. The addresses change during the HIGH phase before the transfer to which they refer, and remain valid until the last HIGH phase of the transfer.
BCLK Bus Clock	In	System (bus) clock. This clock times all bus transfers. The clock has two distinct phases: phase 1 in which **BCLK** is LOW, and phase 2 in which **BCLK** is HIGH.
BERROR Error Response	InOut	A transfer error is indicated by the selected bus slave or decoder using the **BERROR** signal. When **BERROR** is HIGH, a transfer error has occurred. When **BERROR** is LOW, the transfer is successful. This signal is also used in combination with the **BLAST** signal to indicate a bus retract operation. When no bus transfer is taking place, this signal is driven by the decoder. This signal is driven in the LOW phase of **BCLK** and is valid before the rising edge of **BCLK**.
BLAST Last Response	InOut	This signal is driven by the selected bus slave or decoder to indicate whether the current transfer should be the last of a burst sequence. When **BLAST** is HIGH, the next bus transfer must allow sufficient time for address decoding. When **BLAST** is LOW, the next transfer may continue a burst sequence. This signal is also used in combination with the **BERROR** signal to indicate a bus retract operation. When no bus transfer is taking place, this signal is driven by the decoder. This signal is driven in the LOW phase of **BCLK** and is valid before the rising edge of **BCLK**.
BnRES Reset Status	In	This active LOW signal indicates the reset status of the bus and is driven by the reset controller.
BTRAN[1:0] Transfer Type	In	These signals indicate the type of the next transaction, which may be address-only, nonsequential or sequential. These signals are valid during the phase HIGH before the transfer to which they refer.
BWAIT Wait Response	InOut	This signal is driven by the selected bus slave or decoder to indicate whether the current transfer may complete. If **BWAIT** is HIGH, a further bus cycle is required. If **BWAIT** is LOW, the transfer may complete in the current bus cycle. When no bus transfer is taking place, this signal is driven by the bus decoder (this block). This signal is driven in the LOW phase of **BCLK** and is valid before the rising edge of **BCLK**.

Table 1-1 Signal descriptions for the decoder block (continued)

Name	Type	Description
DSELx Slave Select	Out	This signal indicates that a slave device is selected and a data transfer is required. There is a **DSELx** signal for each bus slave. This signal becomes valid during the phase HIGH before the data transfer is required, and remains active until the last phase HIGH of the transfer.
BSIZE[1:0 Transfer Size	In	These signals indicate the size of the transfer which may be byte, halfword or word. They are only needed by the decoder with decode cycles implementation. They have the same timing as the system address bus.
ReMap	In	When LOW, this forces the internal memory to be replaced by the external memory. The external memory should contain the system's start-up program (boot ROM/BIOS). In normal operation this signal is HIGH.

1.3 Memory map

The decoder controls the memory map of the system, and generates a slave select signal for each memory region.

The **ReMap** signal is used to provide a different memory map at reset, when ROM is required at address 0, and during normal operation, when RAM may be used at address 0.

The **ReMap** signal is typically provided by a remap and pause peripheral, which drives **ReMap** LOW at reset. The signal is only driven HIGH after a particular address in the remap and pause peripheral is accessed.

Figure 1-2 shows both the Normal and Reset memory maps.

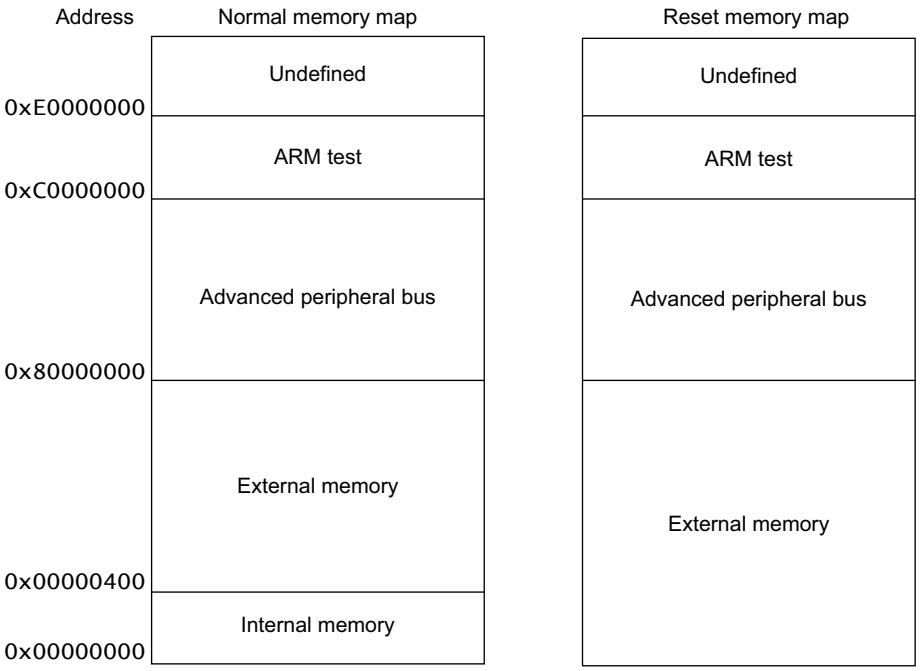


Figure 1-2 Memory map

1.4 Decoder function

At the start of every transfer on the bus the decoder can perform a number of actions. The decoder is able to determine when a transfer is about to start by examining the **BWAIT** signal, which will be LOW when the previous transfer is completing.

The actions the decoder may take depend on the type of transfer:

- Nonsequential transfer

  The decoder inserts a single decode cycle to allow the address bus to stabilize and the new address to be decoded. For the first cycle, the decoder drives **BWAIT** HIGH and negates all **DSELx** signals. In the second cycle, the decoder asserts the appropriate **DSELx**, and the selected slave becomes responsible for driving the slave transfer response. This is the case for a decoder that implements decode cycles, otherwise the decoder asserts the **DSELx** signal during the first cycle (see Figure 1-4 on page 1-8).

- Sequential transfer with **BLAST** LOW

  The decoder drives the appropriate **DSELx** signal, and the selected slave is responsible for driving the slave transfer response.

- Sequential transfer with **BLAST** HIGH

  The decoder treats this case in the same way as a nonsequential transfer.

- Address-only transfer

  The decoder does not generate any **DSELx** signals and drives a slave transfer response of **BWAIT** LOW.

Figure 1-3 on page 1-8 illustrates a decode cycle.

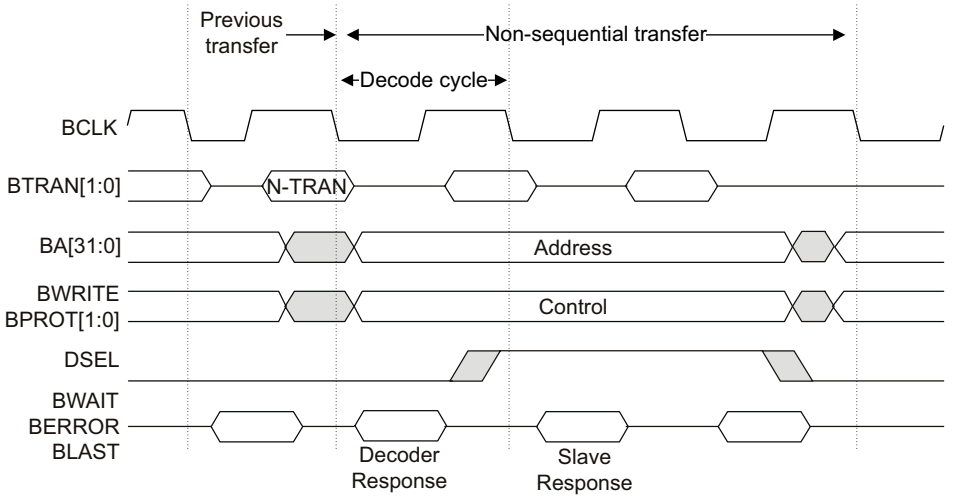


Figure 1-3 Decoder with decode cycles

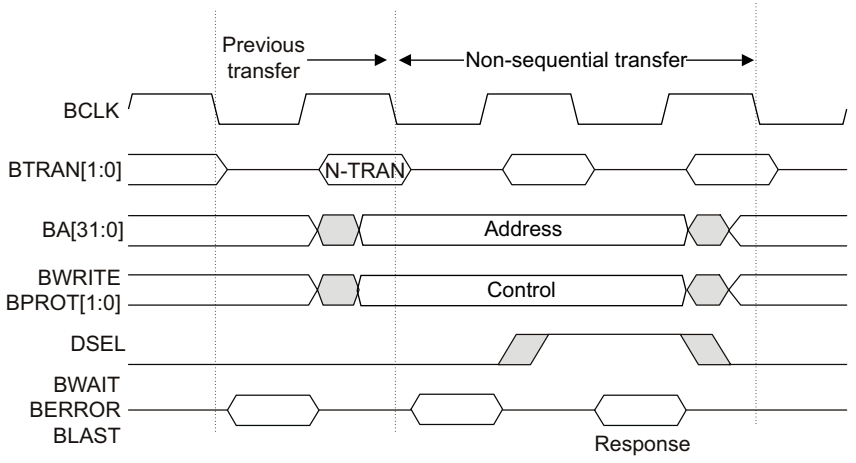


Figure 1-4 Decoder without decode cycles

1.4.1 Slave response

The decoder control block monitors bus activity, and determines when a transfer is required to a slave device. When a transfer is required, a signal is asserted to the address decoder block, which in turn generates a slave select signal depending on the address on the bus.

When no slave is selected, the decoder must provide the slave response signals (**BWAIT**, **BLAST** and **BERROR**) in order for the bus to remain synchronised. The decoder will drive these signals during the address-only cycle (as indicated by the **BTRAN[1:0]** signals which are driven by the bus master), and the decode cycle, which occurs at the start of every nonsequential transfer and may also be requested by a slave device using the **BLAST** signal.

The decoder with decode cycles will provide the following responses:

Table 1-2 Decoder with decode cycles response combinations

Condition	BWAIT	BLAST	BERROR
Address-only cycle	0	0	0
Error	0	0	1
Decode cycle	1	0	0

The decoder without decode cycles does not provide the decode cycle response.

When a slave is selected it must provide a response on the **BWAIT**, **BERROR** and **BLAST** signals.

1.4.2 Implementation of DecError and DecLast

These two internal decoder signals are required by the state machine. **DecError** is used by both state machines, with and without decode cycles. It is asserted when an access is made to the undefined region.

DecLast is only needed by the state machine with decode cycles. It is asserted when the last address in a 1KB boundary is accessed, and is dependent on the size of the transfer.

1.4.3 Decode control state machine

There are two possible implementations of the decoder, depending on the performance requirements of the system design. One implementation inserts decode cycles, and the other does not. The state machine for the decoder with decode cycles has an extra state, DECODE.

The decoder state machine is clocked off the falling edge of the bus clock, **BCLK**. Therefore it is necessary for the decoder to use latched versions of the **BWAIT** and **BLAST** signals. WAIT + RETNEXT is equivalent to **LBWAIT**.

The decoder will drive an address-only cycle response when in the ADDRONLY state, a Decode cycle response when in the DECODE state, and an Error transfer response when in the ERROR state. The decoder will not drive the slave response signals in the SLAVESEL state; this is the responsibility of the selected slave.

Figure 1-5 on page 1-11 shows the state machine of the decoder control block.

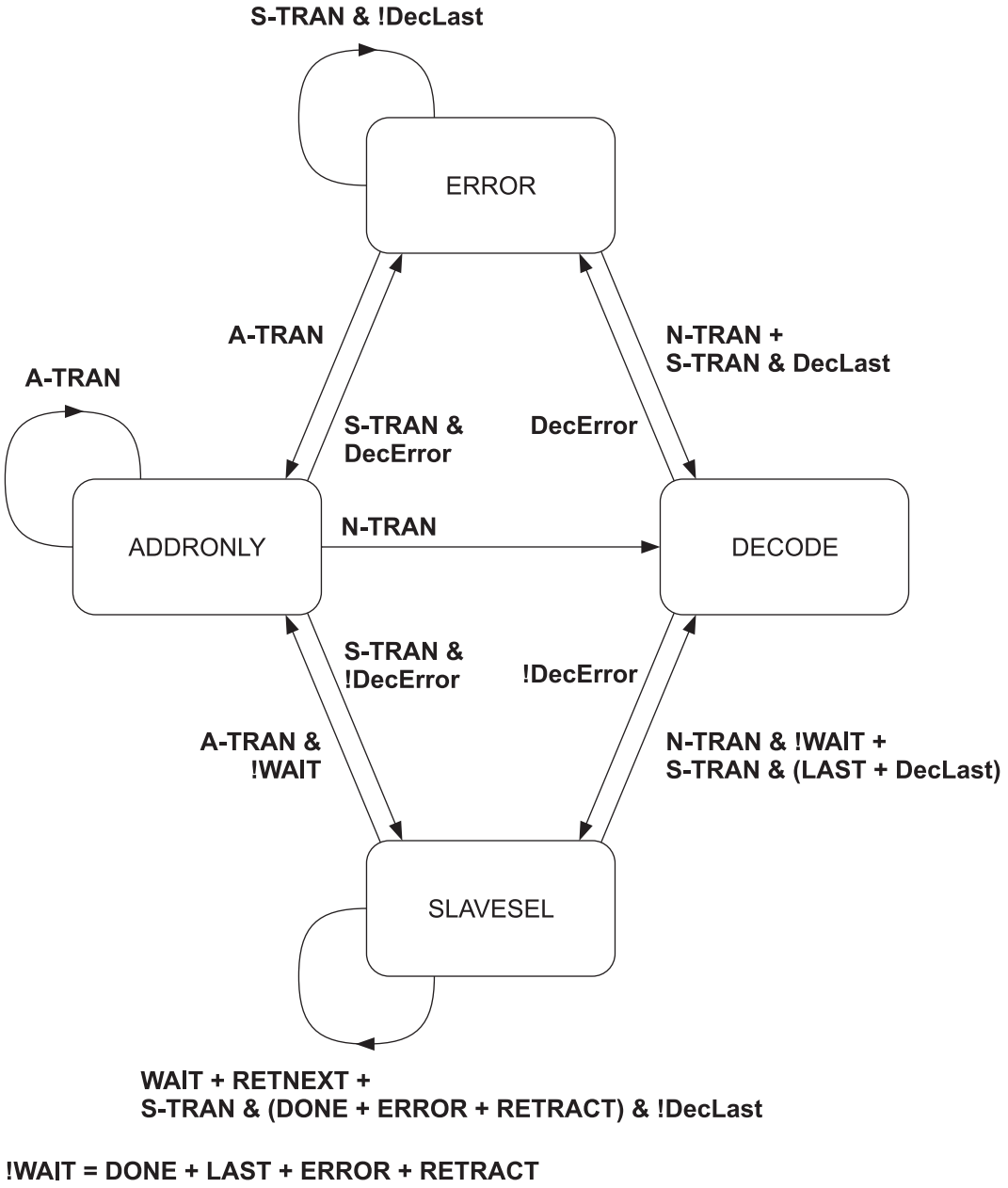


Figure 1-5 Decoder state machine with decode cycles

State descriptions

DECODE Decode cycle. The DECODE state is only entered for a single cycle. Assuming that there is no reset condition and that **DecError** is not asserted, the next state is SLAVESEL.

SLAVESEL Slave transfer. The SLAVESEL state may be entered either from the DECODE or the ADDRONLY state.

Unless there is a reset condition, the state machine will remain in the SLAVESEL state when **BWAIT** is HIGH. When **BWAIT** is LOW, the next state is ADDRONLY if the transfer type is address-only, DECODE if the transfer type is nonsequential, or if **BLAST** or **DecLast** have been asserted. However, if the transfer type is sequential and **BLAST** or **DecLast** have not been asserted, the next state remains as SLAVESEL.

Error Error transfer. The ERROR state is entered when **DecError** is asserted.

Unless there is a reset condition, the state machine remains in the ERROR state when the transfer type is sequential and DecLast is not asserted. The next state is ADDRONLY if the transfer type is address-only, or DECODE if the transfer type is nonsequential, or if it is sequential and **DecLast** is asserted.

Figure 1-6 on page 1-13 shows the state machine of the decoder control block.

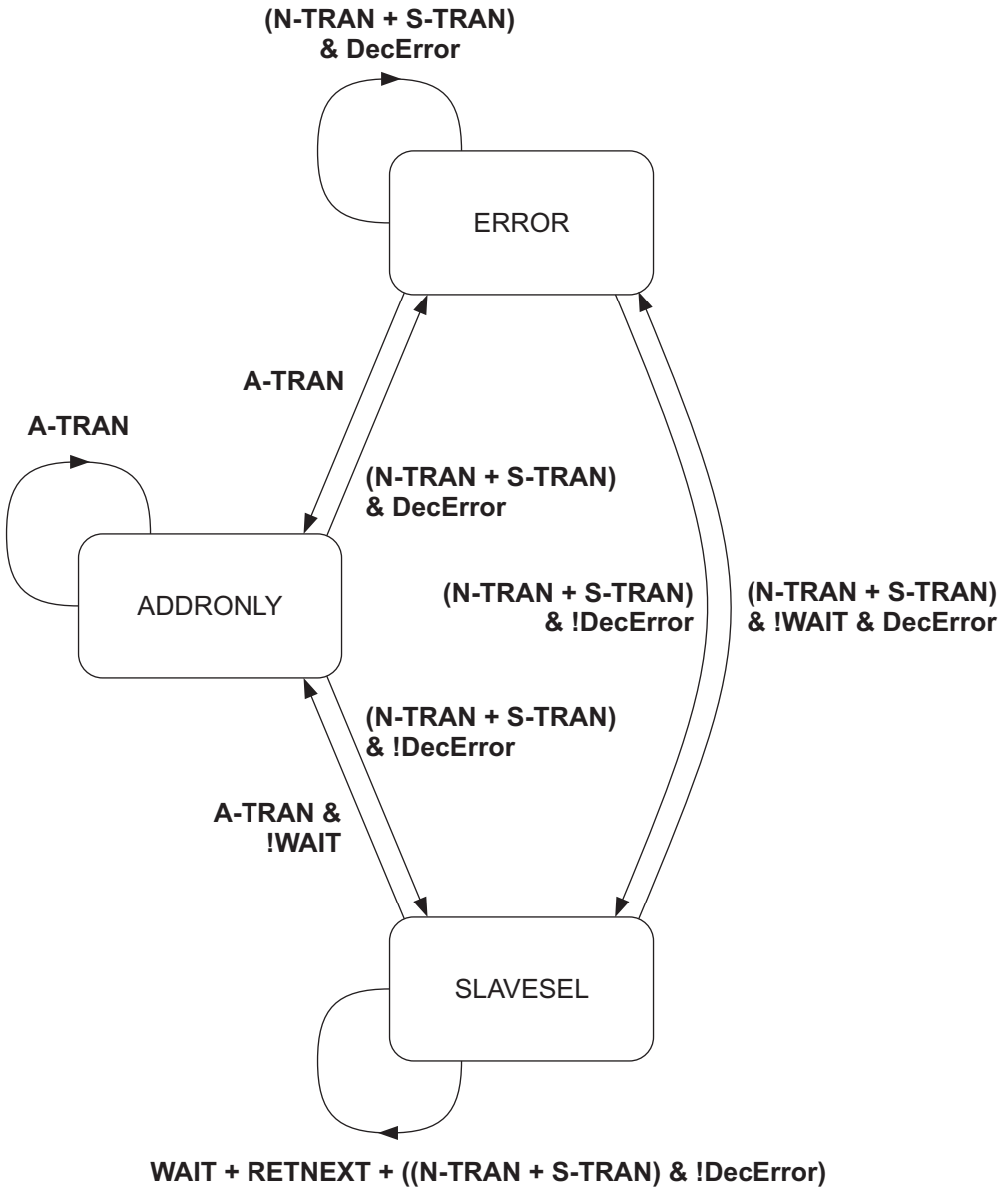


Figure 1-6 Decoder state machine without decode cycles

State descriptions

ADDRONLY No transfer. This state is entered when no data transfer is occurring on the bus. This state is also entered during a reset condition.

The ADDRONLY state is only exited when a transfer is about to occur, as indicated by the **BTRAN** signals. If **DecError** is asserted, the next state is ERROR; if it is not asserted, the next state is SLAVESEL.

SLAVESEL Slave transfer. The SLAVESEL state may be entered from either the ERROR or the ADDRONLY state.

Unless there is a reset condition, the state machine will remain in the SLAVESEL state when **BWAIT** is HIGH. When **BWAIT** is LOW, the next state is ADDRONLY if the transfer type is address-only, or ERROR if **DecError** has been asserted. However, if **DecError** has not been asserted, the next state remains as SLAVESEL.

Error Error transfer. The ERROR state is entered when **DecError** is asserted.

Unless there is a reset condition, the state machine remains in the ERROR state when the transfer type is sequential or nonsequential, and **DecError** is asserted. The next state is ADDRONLY if the transfer type is address-only, or SLAVESEL if the transfer type is sequential or nonsequential and **DecError** is not asserted.

1.4.4 Address decoder

When the decoder control section indicates that a slave transfer is required, the address decoder generates a slave select signal, depending on the address on the bus.

The address decoder section must be changed to implement a different memory map.

Signals, other than the address bus, may be used to affect the address map; **ReMap** is an example of such a signal.

1.4.5 Reset operation

During a reset condition, when **BnRES** is LOW, the decoder control block asynchronously removes all slave select signals.