16-Bit Flow-Through EDAC
Error Detection And Correction unit

Description

The 29C516E MHS EDAC is a very low power flow-through 16-bit Error Detection And Correction unit (EDAC) with two user data buses. The EDAC is used in a high integrity system for monitoring and correction of data values coming from the memory space. During a processor write cycle, at each memory location (16-bit width), EDAC calculated checkword (6 or 8-bit width) is added. When performing a read operation from memory, the 29C516E verifies the entire checkword and data combination. It detects and can correct 100% of all the single-bit errors and it detects all double-bit errors. When the 29C516E uses 6-checkbit, it can detect any error on any single 4-bit memory chip. The 8-check-bit option gives the additional capability to detect all errors on any single 8-bit memory chip. All the errors are signaled to the master system (via 2 error Flags) in order to allow the processor to make the required action.

The 29C516E operates in two possible modes: corrected or detected mode. In the corrected mode, the single-bit in error is complemented (corrected). Then, the available entire data is placed on the output port and the Correctable Error Flag is set. In case of double-bit errors (or more), the corrupted data is placed on the output port and the Uncorrectable Error Flag is set. Note that when there is more than two errors, then some bit patterns may appear as possible correctable errors. Therefore, if the environment produces this type of error, the EDAC must be used in detect and provide no automatic correction. Data and syndrome analysis must be done.

The 29C516E acts as a data buffer for µP-memory interfacing. A flow-through EDAC is placed in the data bus path, between the processor and the memory to be protected. This component is able to serve two different users of one memory space. So, it forms the interface between the 22/24-bit (16+6/16+8) memory data bus and the two 16-bit processor data busses with a high drive capability (-12.8 mA). The two data ports can be used to create a dual port bus in front of memory space. The User-1(2) can transfer data from/to the memory or from/to the User-2(1), by-passing the memory. During read or write memory cycles processed by the User-1(2), the User-2(1) have the possibility to listen the transferred data.

Features

- Very Low Power CMOS
- 16-Bit operation with 6 or 8 Check Bits
- Fast Error Detection : 31 ns (max.)
- Fast Error Correction : 32 ns (max.)
- Corrects all Single-Bit Errors
  Detects all Double-Bit Errors
  Detects some Multi-Bit Errors
  Detects Chip Errors (x1, x4 & x8 RAM Format)
- Correctable and Uncorrectable Error Flags
- Two User Data Buses
- User to User Transfer and Listening operation
- High Drive Capability on Buses : -12.8 mA
- TTL Compatible
- Single 5V ±10% Power Supply
- 100 Pin Multilayer Quad Flat Pack (Flat leaded or L leaded).

Interface

Functional Diagram

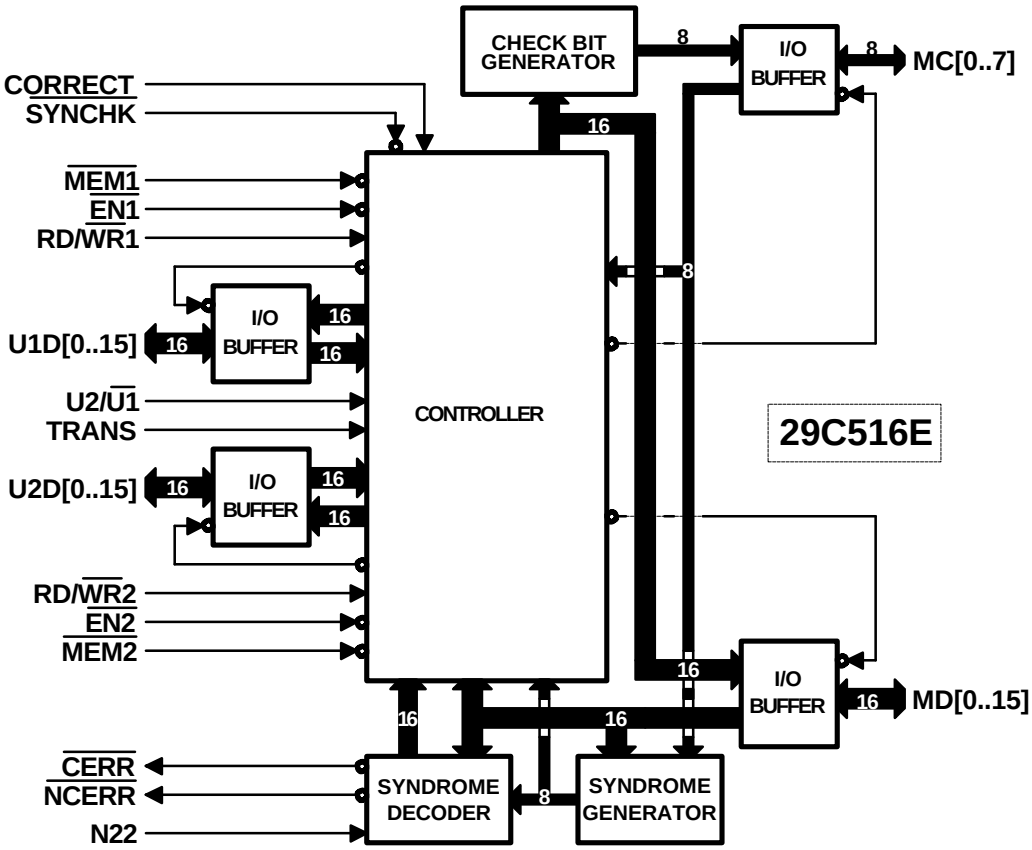


Figure 1. Functional Diagram

Block Diagram

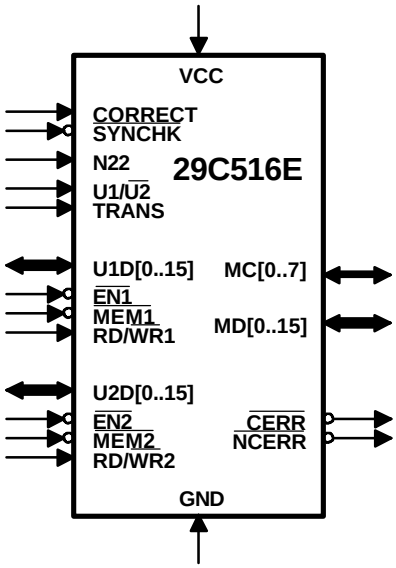


Figure 2. Block Diagram

Pin Configuration for multilayer quad Flat-pack (flat or L leaded)

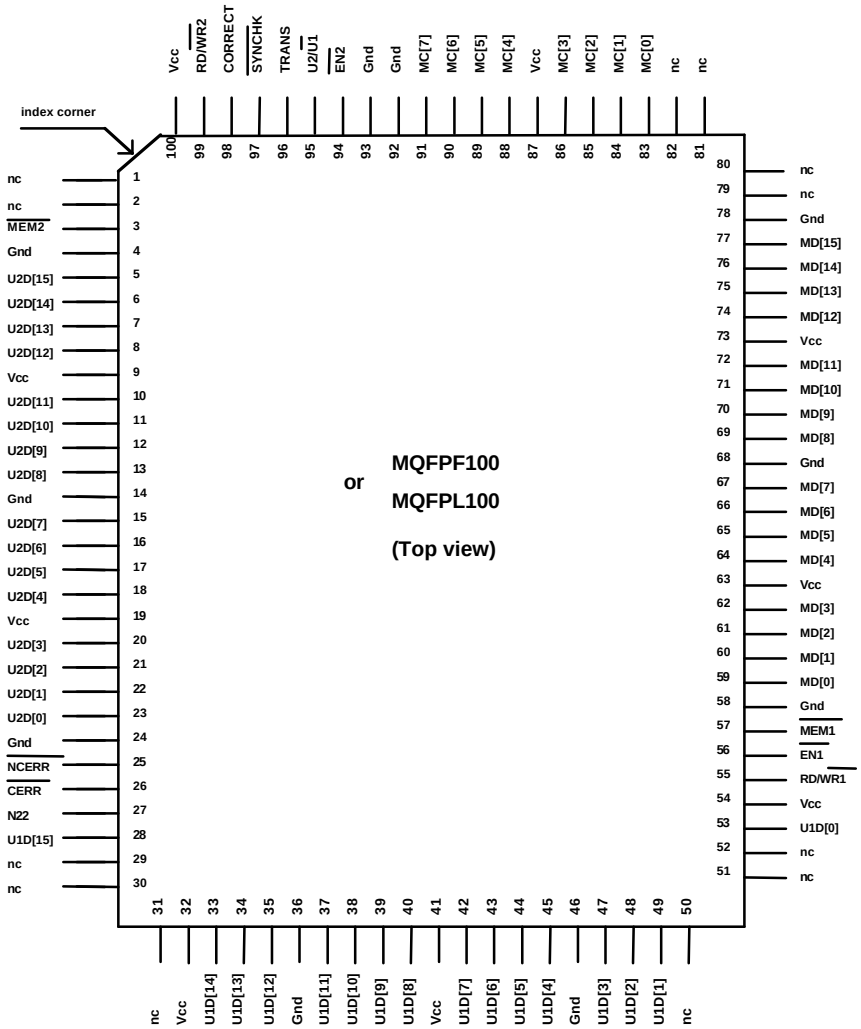


Figure 3. Pin Configuration

Pin Description

Table 1.

Name	Pin	I/O	Active	Description
Buses				
U1D[0..15]	53,49..47,45..42,40..37,35..33,28	I/O *	High	User 1 Data Bus
U2D[0..15]	23..20,18..15,13..10,8..5	I/O *	High	User 2 Data Bus
MD[0..15]	59..62,64..67,69..72,74..77	I/O *	High	Memory Data Bus
MC[0..7]	83..86,88..91	I/O *	High	Memory Check-bit Bus
Error Flags				
$\overline{\text{CERR}}$	26	O	Low	Correctable Error
$\overline{\text{NCERR}}$	25	O	Low	Uncorrectable Error
General Control Signals				
CORRECT	98	I *	High	When active, the EDAC is in CORRECT mode. If low, the EDAC is in DETECT mode.
$\overline{\text{SYNCHK}}$	97	I *	Low	Selects the Syndrome bits (high byte) and the Check-bits (low byte) to be driven on the selected User Data Bus.
N22	27	I *	High	When active, the EDAC uses 6 check-bits. If low, the EDAC uses 8 check-bits in memory read.
TRANS	96	I *	H/L	Selects the Data path to be used. If high, the EDAC access the memory, if low, the EDAC access the transfer buffer.
U2/$\overline{\text{U1}}$	95	I *	H/L	Selects who is the master of User 1 and User 2. The master is responsible for applying RD/$\overline{\text{WR}}$x, $\overline{\text{MEM}}$x, and $\overline{\text{EN}}$x signals in a correct way.
User 1 Control Signals				
RD/$\overline{\text{WR1}}$	55	I *	H/L	User 1 Read/Write signal
$\overline{\text{EN1}}$	56	I *	Low	User 1 Output Enable
$\overline{\text{MEM1}}$	57	I *	Low	User 1 Memory Select
User 1 Control Signals				
RD/$\overline{\text{WR2}}$	99	I *	H/L	User 2 Read/Write signal
$\overline{\text{EN2}}$	94	I *	Low	User 2 Output Enable
$\overline{\text{MEM2}}$	3	I *	Low	User 2 Memory Select
Power (Buffers)				
VCCB	9,19,32,41,54,63,73,87	I	-	Buffers supply (5 V nominal)
GNDB	4,14,24,36,46,58,68,78,92	I	-	Buffers 0 V reference
Power (Core)				
VCCC	100	I	-	Core supply (5 V nominal)
GNDC	93	I	-	Core 0 V reference

* Pull-up buffers

Check-Bit Generation

The Check-bit Generator produces 8 check-bits (whatever N22 value) from the incoming User Data Word UxD[0..15] according the Table 2.

Example: to create check-bit 0, bit 13, 12, 8, 7, 6, 5, 4 and 0 of the Data Word are XORed together.

If memory devices 8-bit wide are used, 24 bits (MD[0..15] & MC[0..7]) are stored to give error detection. But if memory devices 1-bit or 4-bit wide are used, 22 bits (MD[0..15] & MC[0..5]) are stored to give error detection.

Table 2. Check Bit Generation (√ indicates a bit of UxD bus used in the XOR/NXOR)

MC[..]	PARITY	UxD[..]															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Even (XOR)			√	√				√	√	√	√	√				√
1	Even (XOR)		√			√	√	√	√		√		√			√	
2	Odd (NXOR)	√			√			√				√		√	√	√	√
3	Odd (NXOR)		√	√			√						√	√	√	√	√
4	Even (XOR)	√				√	√	√	√	√		√			√		
5	Even (XOR)	√	√	√	√	√				√	√			√			
6	Even (XOR)			√		√			√	√		√	√		√	√	
7	Odd (NXOR)	√		√	√	√						√		√	√	√	

Syndrome Generation

The syndrome Generator produces 8 syndrome-bits (whatever N22 value) from the incoming Memory Data Word MD[0..15] and the associated Check-bits MC[0..7] (or MC[0..5]) according the Table 3.

Syndrome-bit SY[x] is the XOR of the generated Check-bit MC[x] with the generation of Chek-bit on MD[..].

Example: to create syndrome-bit 3, first the bit 14, 13, 10, 4, 3, 2, 1 and 0 of the Data Word (MD[14,13,10,4,3,2,1,0]) are NXORed. Then, the result is XORed with the associated Check-bit (MC[3]) of the Check-byte read in the same time as Data Word is checked.
If the memory uses x8 devices, then the bits should be physically divided as follows: MC[0..7], MD[0..7] and MD[8..15] . For x4 organisation, the bits should be divided MC[0..2]+MC[6], MC[3..5]+MC[7], MD[0..3], MD[4..7], MD[8..11] and MD[12..15].

Table 3. Syndrome Bit Generation (√ indicates a bit of MD and MC buses used in the XOR/NXOR)

SY[..]	PARITY	MD[..]																MC[..]							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	7	5	4	3	6	2	1	0
0	Even (XOR)			√	√				√	√	√	√	√				√								√
1	Even (XOR)		√			√	√	√	√		√		√			√							√		
2	Odd (NXOR)	√			√			√				√		√	√	√	√						√		
3	Odd (NXOR)		√	√			√						√	√	√	√	√				√				
4	Even (XOR)	√				√	√	√	√	√		√			√					√					
5	Even (XOR)	√	√	√	√	√				√	√			√					√						
6	Even (XOR)			√		√			√	√		√	√		√	√						√			
7	Odd (NXOR)	√		√	√	√						√		√	√	√		√							

Syndrome Decoding

The syndrome decoder generates the error flags /CERR (Correctable ERRor) and /NCERR (Non-Correctable ERRor). If a correctable error occurs, the 29C516E EDAC provides corrected data to the user. The inputs are the 8 syndrome bits from the syndrome generator, the 16 data bits from the memory and the control signal N22. N22 signal controls if 22 or 24 bits shall be decode from the entire memory word.

Table 4. 6-Bit Syndrome Word to Bit-In-Error (N22="1")

Syndrome Bit SY[..]				Hex	0	1	2	3
				5	0	0	1	1
				4	0	1	0	1
Hex	3	2	1	0				
0	0	0	0	0	N.E.D	MC4	MC5	D
1	0	0	0	1	MC0	D	D	MD7
2	0	0	1	0	MC1	D	D	MD11
3	0	0	1	1	D	MD8	MD6	D
4	0	1	0	0	MC2	D	D	MD15
5	0	1	0	1	D	MD5	MD12	D
6	0	1	1	0	D	MD9	M	D
7	0	1	1	1	M	D	D	M
8	1	0	0	0	MC3	D	D	M
9	1	0	0	1	D	M	MD13	D
A	1	0	1	0	D	MD10	MD14	D
B	1	0	1	1	MD4	D	D	M
C	1	1	0	0	D	MD2	MD3	D
D	1	1	0	1	MD0	D	D	M
E	1	1	1	0	MD1	D	D	M
F	1	1	1	1	D	M	M	D

Note: **N.E.D = No Errors Detected**
MDx = Memory Data Bit-In-Error
MCx = Memory Check Bit-In-Error
D **= Double-Bit-In-Error Detected**
M **= Multi-Bit-In-Error Detected**

Table 5. 8-Bit Syndrome Word to Bit-In-Error (N22="0")

Syndrome Bit SY[..]				Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
				7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				6	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				5	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Hex	3	2	1	0																
0	0	0	0	0	N.E.D	MC4	MC5	D	MC6	D	D	D	MC7	D	D	D	D	M	M	D
1	0	0	0	1	MC0	D	D	D	D	D	D	MD7	D	M	D	M	M	D	D	D
2	0	0	1	0	MC1	D	D	M	D	D	D	D	D	M	D	D	M	D	D	MD11
3	0	0	1	1	D	D	MD6	D	D	MD8	D	D	D	M	D	D	D	D	M	D
4	0	1	0	0	MC2	D	D	D	D	M	M	M	D	D	D	MD15	M	D	D	D
5	0	1	0	1	D	M	D	D	D	D	M	M	D	D	MD12	D	D	MD5	D	D
6	0	1	1	0	D	MD9	M	D	D	D	M	M	D	D	M	D	D	M	M	D
7	0	1	1	1	M	D	D	M	M	D	D	D	M	D	D	M	M	D	D	M
8	1	0	0	0	MC3	D	D	M	D	M	D	D	D	D	D	M	M	D	D	M
9	1	0	0	1	D	M	M	D	D	M	D	D	M	M	D	D	D	M	MD13	D
A	1	0	1	0	D	MD10	MD14	D	D	D	D	D	M	D	D	D	D	M	M	D
B	1	0	1	1	D	D	D	M	MD4	D	D	D	M	M	M	M	D	M	D	M
C	1	1	0	0	D	M	D	D	D	D	M	M	D	D	MD3	D	D	MD2	D	D
D	1	1	0	1	MD0	D	D	M	D	D	M	M	D	D	D	M	M	D	D	M
E	1	1	1	0	M	D	D	M	D	D	M	M	D	D	D	M	MD1	D	D	M
F	1	1	1	1	D	M	M	D	D	M	M	M	D	M	M	D	D	M	M	D

The 6-Bit Syndrome Word

This feature is available when the N22 pin is driven at a high level.

No Errors

If there are no errors in the read Data or Check-Bit, all the syndrome byte is **"00"**. The EDAC flags are inactive.

No Error : **SY=00**

Single Bit-Error

A single bit-error in a Memory Data word read (MD[..]) causes three syndrome bits to be set to one. The code formed indicates which bit of the Memory Data word is incorrect.

For example, if MD[2] were incorrect, the syndrome byte would have bits 2, 3 and 4 set to one. The syndrome decoder of 29C516E EDAC decodes the information in the syndrome byte and only sets low the error flag $\overline{CERR}$. In correct mode (CORRECT pin active), it inverts (and hence corrects) the relevant bit in error of the Memory Data word and provides the expected Data word for the EDAC controller.

If there is an error in the Memory Check-bit (MC[..]), only one bit of the syndrome is set to one.

In this case, the syndrome decoder sets low the correctable error flag $\overline{CERR}$, but $\overline{NCERR}$ does not change. It does not correct the Check-bit because these bits are not used by the system.

Table 6. Single Bit Error

MD[..]	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
$SY_{(hexa)}$	34_h	$2A_h$	29_h	25_h	32_h	$1A_h$	16_h	13_h	31_h	23_h	15_h	$0B_h$	$2C_h$	$1C_h$	$0E_h$	$0D_h$

MC[..]	[-]	[-]	[5]	[4]	[3]	[2]	[1]	[0]
$SY_{(hexa)}$	$--_h$	$--_h$	20_h	10_h	08_h	04_h	02_h	01_h

Double-Bit Error

If two errors occurs, there will be either 2, 4 or 6 bits set to one in the syndrome byte. The syndrome value generated by a double-bit error does not take place of a syndrome value generated by a single-bit error. Then, only the non correctable error flag $\overline{NCERR}$ will be activated to indicate that errors are present but cannot be corrected.
Example: If MD[4] and MC[2] are incorrect, syndrome bits [0], [1], [2] and [3] are set to one ($SY=0F_h$), $\overline{NCERR}$ is set low and $\overline{CERR}$ remains at high level.

Triple-Bit Error

When three errors are detected, an error flag is set low as warning to the system. But the generated syndrome can have the listed value of single-bit error. The device must be in detect mode to prevent false correction occurring.

Example: If MD[0], MD[14] and MC[1] are corrupted, the syndrome vaue is "25_h ". This is decoded by the 29C516E EDAC as being a correctable error on MD[12]. The $\overline{CERR}$ flag is set low and correction would take place if the device is in correct mode. This would cause more errors.

4-bit Wide Memory Error

The 6 check-bit code can be used to provide error detection for up to 4 errors occuring in the following groups: MD[15..12], MD[11..8], MD[7..4], MD[3..0], MC[5..3] and MC[2..0]. The 29C516E EDAC can flag any number of errors in 4-bit wide memory chip.

A special attention must be taken, muti-bit error (≥3) located into the defined groups can provide the syndrome byte of a single-bit error.
Example: If MD[3], MD[2], MD[1] and MD[0] are in error, the syndrome code is "33_h";

The 8-Bit Syndrome Word

This feature is available when the N22 pin is driven at a low level.

No Errors

If there are no errors in the read Data or Check-Bit, all the syndrome byte is **"00"**. The EDAC flags are inactive.

No Error : **SY=00**

Single Bit-Error

A single bit-error in a Memory Data word read (MD[..]) causes three syndrome bits to be set to one. The code formed indicates which bit of the Memory Data word is incorrect.

For example, if MD[10] were incorrect, the syndrome byte would have bits 1, 3 and 4 set to one. The syndrome decoder of 29C516E EDAC decodes the information in the syndrome byte and only sets low the error flag $\overline{CERR}$. In correct mode (CORRECT pin active), it inverts (and hence corrects) the relevant bit in error of the Memory Data word and provides the expected Data word for the EDAC controller.

If there is an error in the Memory Check-bit (MC[..]), only one bit of the syndrome is set to one.

In this case, the syndrome decoder sets low the correctable error flag $\overline{CERR}$, but $\overline{NCERR}$ does not change. It does not correct the Check-bit because these bits are not used by the system.

Table 7. Single Bit Error

MD[..]	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
$SY_{(hexa)}$	$B4_h$	$2A_h$	$E9_h$	$A5_h$	$F2_h$	$1A_h$	16_h	53_h	71_h	23_h	$D5_h$	$4B_h$	AC_h	DC_h	CE_h	$0D_h$

MC[..]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
$SY_{(hexa)}$	80_h	40_h	20_h	10_h	08_h	04_h	02_h	01_h

Double-Bit Error

If two errors occur, there will be 2, 3, 4, 5, 6 or 8 bits set to one in the syndrome byte. The syndrome value generated by a double-bit error does not take place of a syndrome value generated by a single-bit error. Then, only the non correctable error flag $\overline{\text{NCERR}}$ will be activated to indicate that errors are present but cannot be corrected.

Example: If MD[5] and MC[7] are incorrect, syndrome bits [0], [2], [4] and [6] are set to one ($SY=55_h$), $\overline{\text{NCERR}}$ is set low and $\overline{\text{CERR}}$ remains at high level.

Triple-Bit Error

When three errors are detected, an error flag is set low as warning to the system. But the generated syndrome can have the listed value of single-bit error. The device must be in detect mode to prevent false correction occurrence.
Example: If MD[0], MD[9] and MC[0] are corrupted, the syndrome value is "$1A_h$".

This is decoded by the 29C516E EDAC as being a correctable error on MD[10]. The $\overline{\text{CERR}}$flag is set low and correction would take place if the device is in correct mode. This would cause more errors.

4-bit Wide Memory Error

The 8 check-bit code can be used to provide error detection for up to 4 errors occur in the following groups: MD[15..12], MD[11..8], MD[7..4], MD[3..0], MC[7..4] and MC[3..0]. The 29C516E EDAC can flag any number of errors in 4-bit wide memory chip.

A special attention must be taken, multi-bit error (≥3) located into the defined groups can provide the syndrome byte of a single-bit error.
Example: If MD[11], MD[10], MD[9] and MD[8] are in error, the syndrome code is "AD_h".

8-bit Wide Memory Error

The 8 check-bit code can be used to provide error detection for up to 8 errors occuring in the following groups: MD[15..8], MD[7..0] and MC[7..0].
The 29C516E EDAC can flag any number of errors in 8-bit wide memory chip. A special attention must be taken, muti-bit error (≥3) located into the defined groups can provide the syndrome byte of a single-bit error.

Example: If MD[13], MD[12], MD[10] and MD[9] are in error, the syndrome code is "40_h". (In 6 check-bit coding, the syndrome code should have been "00_h", the "No Error Detected" value.) Note that the syndrome code "40_h" is also the code for MC[6] in error.

Transactions

Three types of transactions may be done:

Memory Read

The TRANS pin is driven at a high level to select the access to the memory. The external arbiter drives the U2/U1 pin and dispatches the commands RD$\overline{WRx}$, $\overline{MEMx}$ and $\overline{ENx}$. All transaction managed by the master user can be listened by the second user.

Table 8.

TRANS	U2/$\overline{U1}$	CORRECT	$\overline{SYNCHK}$	RD/$\overline{WR1}$	$\overline{EN1}$	$\overline{MEM1}$	RD/$\overline{WR2}$	$\overline{EN2}$	$\overline{MEM2}$	$\overline{CERR}$	$\overline{NCERR}$	Function
1	0	1	1	1	0	0	x	x	x	1	1	UD1[0..15] = MD[0..15]
										0	1	UD1[0..15] = {corrected MD[0..15]}
										x	0	UD1[0..15] = {corrupted MD[0..15]}
		0	1	1	0	0	x	x	x	x	x	UD1[0..15] = MD[0..15]
		x	0									UD1[0..15] = {MC[0..7] | Syndrome}
		x	x	1	1	x	x	x	x	x	x	UD1[0..15] = H.Z
					x	1						
				1	x	x	1	0	0	x	x	UD2[0..15] = {expected UD1[0..15]} (User 2 listening)
1	1	1	1	x	x	x	1	0	0	1	1	UD2[0..15] = MD[0..15]
										0	1	UD2[0..15] = {corrected MD[0..15]}
										x	0	UD2[0..15] = {corrupted MD[0..15]}
		0	1	x	x	x	1	0	0	x	x	UD2[0..15] = MD[0..15]
		x	0									UD2[0..15] = {MC[0..7] | Syndrome}
		x	x	x	x	x	1	1	x	x	x	UD2[0..15] = H.Z
								x	1			
				1	0	0	1	x	x	x	x	UD1[0..15] = {expected UD2[0..15]} (User 1 listening)

x : don't care

Memory Write

The TRANS pin is driven at a high level to select the access to the memory. The external arbiter drives the U2/U1 pin and dispatches the commands RD$\overline{WR}$x, $\overline{MEM}$x and $\overline{EN}$x. All transaction managed by the master user can be listened by the second user.

Table 9.

TRANS	U2/$\overline{U1}$	RD/$\overline{WR1}$	$\overline{EN1}$	$\overline{MEM1}$	RD/$\overline{WR2}$	$\overline{EN2}$	$\overline{MEM2}$	Function
1	0	0	0	0	x	x	x	MD[0..15] = UD1[0..15] MC[0..7] = {check-bits generated from UD1[0..15]}
		0	1	x	x	x	x	MD[0..15] = H.Z MC[0..7] = H.Z
			x	1				
		0	x	x	1	0	0	UD2[0..15] = UD1[0..15] (User 2 listening)
1	1	x	x	x	0	0	0	MD[0..15] = UD2[0..15] MC[0..7] = {check-bits generated from UD2[0..15]}
		x	x	x	0	1	x	MD[0..15] = H.Z MC[0..7] = H.Z
						x	1	
		1	0	0	0	x	x	UD1[0..15] = UD2[0..15] (User 1 listening)

x : don't care

$\overline{CERR}$ and $\overline{NCERR}$ are not valid

CORRECT and $\overline{SYNCHK}$ are not active

User to User Transfer

The TRANS pin is driven at a low level to select this mode. The external arbiter drives the U2/U1 pin and dispatches the unidirectional commands RD$\overline{WR}$x, $\overline{MEM}$x and $\overline{EN}$x.

Table 10.

TRANS	$\overline{U2}$/U1	RD/$\overline{WR1}$	$\overline{EN1}$	$\overline{MEM1}$	RD/$\overline{WR2}$	$\overline{EN2}$	$\overline{MEM2}$	Function
0	0	1	0	1	x	x	x	UD1[0..15] = UD2[0..15]
			1	x	x	x	x	UD1[0..15] = H.Z
			x	0				
		0	0	1	x	x	x	UD2[0..15] = UD1[0..15]
			1	x	x	x	x	UD2[0..15] = H.Z
			x	0				
0	0	x	x	x	1	0	1	UD2[0..15] = UD1[0..15]
						1	x	UD2[0..15] = H.Z
						x	0	
					0	0	1	UD1[0..15] = UD2[0..15]
						1	x	UD1[0..15] = H.Z
						x	0	

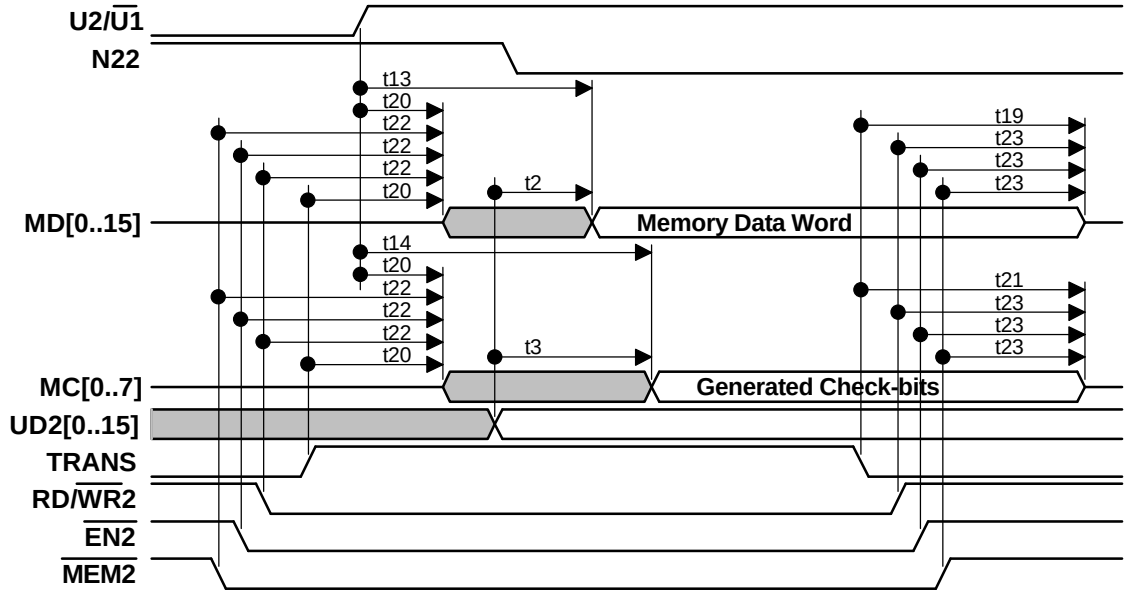
x : don't care

$\overline{CERR}$ and $\overline{NCERR}$ are not valid

CORRECT and $\overline{SYNCHK}$ are not active

Signal Timing

Memory Write



Propagation Delays

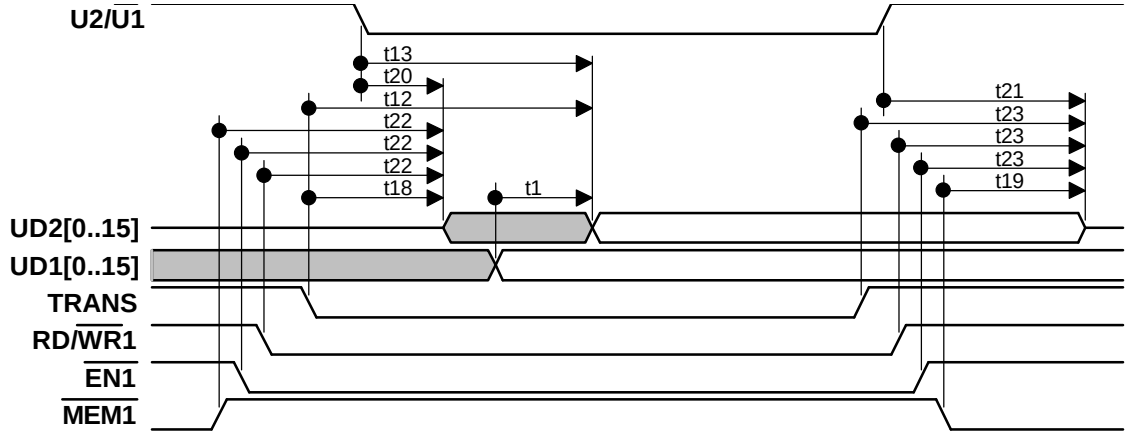
t2 *	t3 *	t13 *	t14 *	(* : Max Value)
13 ns	26 ns	18 ns	30 ns	

Output Enable / Disable Times

t19 *	t20 *	t21 *	t22 *	(* : Max Value)
23 ns	22 ns	22 ns	19 ns	

Figure 4. Memory Write Timing Diagram

Transfer Write



Propagation Delays

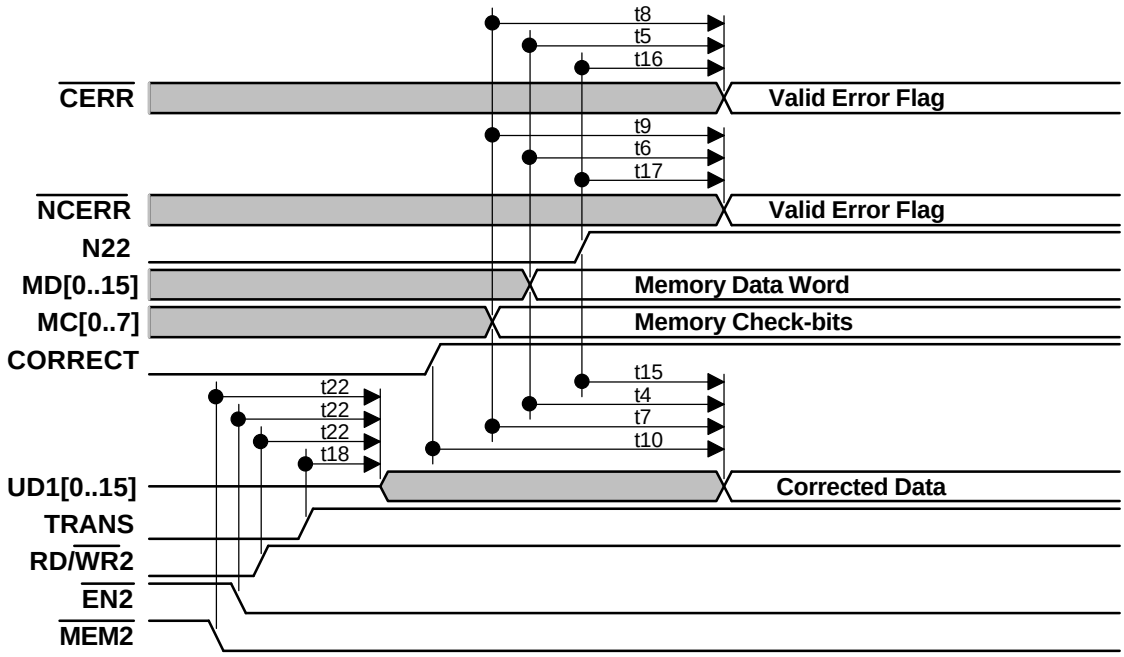
t1 *	t12 *	t13 *	(* : Max Value)
14 ns	20 ns	18 ns	

Output Enable / Disable Times

t18 *	t19 *	t20 *	t21 *	t22 *	t23 *	(* : Max Value)
23 ns	23 ns	22 ns	22 ns	19 ns	19 ns	

Figure 5. Transfer Write Timing Diagram

Memory Read



Propagation Delays

t4 *	t5 *	t6 *	t7 *	t8 *	(* : Max Value)
34 ns	33 ns	34 ns	32 ns	31 ns	

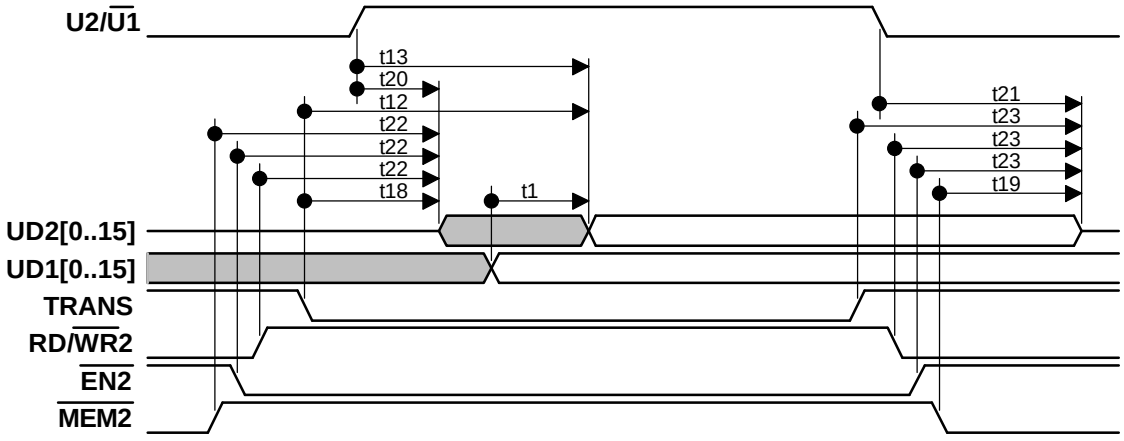
t9 *	t10 *	t15 *	t16 *	t17 *	(* : Max Value)
32 ns	19 ns	24 ns	24 ns	24 ns	

Output Enable / Disable Times

t18 *	t22 *	(* : Max Value)
23 ns	19 ns	

Figure 6. Memory Read Timing Diagram

Transfer Read



Propagation Delays

t1 *	t12 *	t13 *
14 ns	20 ns	18 ns

(* : Max Value)

Output Enable / Disable Times

t18 *	t19 *	t20 *	t21 *	t22 *	t23 *
23 ns	23 ns	22 ns	22 ns	19 ns	19 ns

(* : Max Value)

Figure 7. Transfer Read Timing Diagram

Electrical Characteristics

Absolute Maximum Ratings

Table 11.

Parameter	Value
Supply voltage , Vcc	- 0.5 to 7 V
Input voltage range	- 0.5 to Vcc + 0.5 V
Input current per power pin	+/- 50 mA
Input current per signal pin	+/- 10 mA
Continuous output current, one pin	+/- 30 mA
Soldering lead temperature 1.6 mm from case for max 10 s	+ 300 °C
Storage temperature	- 65 °C to + 150 °C
Maximum package power dissipation	1.0 W

Operating Conditions

Table 12.

Parameter	Min	Typ	Max	Unit
Supply voltage , Vcc	4.5	5.0	5.5	Volt
Operating temperature range	- 55		125	°C

Static Electrical Characteristics

Table 13.

Parameter		Condition	Min	Typ	Max	Unit
VIH	High level input voltage		2,2			V
VIL	Low level input voltage				0,8	V
VOH1	High level output voltage	**I**OH = - 20 µA	Vcc-0.1		V	
VOL1	Low level output voltage	**I**OL = + 20 µA			0,1	V
VOH2	High level output voltage	**I**OH = - 12.8 mA	3,7			V
VOL2	Low level output voltage	**I**OL = + 12.8 mA			0,4	V
IIL	Low level input current	**V**in = **G**nd	- 10	- 1		µA
IILP	Low level input current, (Pull-up Input)	**V**in = **G**nd	- 100	- 40		µA
IIH	High level input current	**V**in = **V**cc		+ 1	+ 10	µA
IIHP	High level input current, (Pull-down Input)	**V**in = **V**cc		+ 40	+ 100	µA
IOZ	Output leakage current	Outputs disable, (**G**nd<**V**out<**V**cc)	- 10		+ 10	µA
IOZLP	Output leakage current, (Pull-up Input)	Outputs disable, (**V**out=**G**nd)	- 100	- 40		µA
IOZHP	Output leakage current, (Pull-down Input)	Outputs disable, (**V**out=**V**cc)		+ 40	+ 100	µA
C1	Input pin capacitance				8	pF
CIO	I/O pin capacitance				12	pF
ICCSB	Standby supply current			+ 10	+ 20	µA

Ordering information

Temperature Range		Package		Device		Speed	Flow
S	**M**	**FR**	-	**29C516E**	-	**31**	**SB**
M : Military S : Space	M : 5V version	KR : MQFPF100 FR : MQFPL100		EDAC16 rad tolerant		31ns	Blank : MHS Military flow /883 : Mil STD 883 class B or S SB : ESA/SCC 9000 level B * SC : ESA/SCC 9000 level C * SLx : LAT parts (1,2 or 3) * * : compliance to ESA/SCC specification to be done.