PROJECT

32-BIT MICROPROCESSOR AND COMPUTER SYSTEM DEVELOPMENT

TITLE

ERC32 SYSTEM OVERVIEW REV. CBA

	Name	Function	Date	Signature
Prepared :	Mikael Ramström	Project Engineer		
Checked :	Bo Törnberg	Design Engineer		
Authorized :	Ritva Svenningsson	Project Manager		

Distribution
Complete : ESTEC/5, B2S, CK-MR, CK-MW, CK-AR, CM-BT, CK-JO, CM-TS, CM-DM, CA
Summary :

Class :		Host System :	Word 2.0c for Windows, SE Macro Rev 1.0B
Contract No :		Host File :	\\svax2\Arkiv1\mcd\tnt\0020_03.doc

SUMMARY

This document is intended as an introduction to system design using the ERC32 processor core chip set, revision CBA.~~This document is intended as an introduction t~~o ~~system design using the ERC32 processor core chip set, revsiion CBA~~.

DOCUMENT CHANGE RECORD

Changes between issues are marked with a left-bar.

Issue	Date	Paragraphs affected	Change information
1	8 May 1996	All	New document
2	3 Feb 1997	1, 2, 3.1, 4.2, 4.3 and Schematic	Updated after ERC32 electrical charaterisaion and for DEM32 V.3
3	10 Apr 1997	3.3.1.1.4, 4.3	Added DPARIO workaround. Updated after MEC specification issue 4.

TABLE OF CONTENTS Page

LIST OF FIGURES

1. SCOPE

This document is intended as an introduction to system design using the ERC32 processor core chip set, revision CBA. The reader is assumed to be familiar with the ERC32 development programme and its objectives. Also, some knowledge about the ERC32 core components (IU, FPU and MEC) is assumed.

This document provides an overview of several different architectures built around the ERC32 core as well as a detailed description of a running demonstration board built around revision CBA of the ERC32 components (IU, FPU and MEC). It does not provide in-depth analyses of the internal functionality of the ERC32 components, rather the principles on how to design a complete system are addressed.

A list of available documents describing ERC32 is provided in paragraph 2. In particular, the *ERC32 System Design Document* provides a detailed description of one computer implementation using the ERC32 core, however this document is a paper design based on an early estimation of the ERC32 performance. The *ERC32 Functional Description* document is more focused on the functions provided by the ERC32 core and serves as overview of the detailed descriptions of the ERC32 components and is fully updated to reflect the current design status of the ERC32 components.

The reader should consult the *IU-RT Preliminary Device Specification* or *TSC691E User´s Manual*, *FPU-RT Preliminary Device Specification* or *TSC692E User´s Manual* and the *MEC Device Specification* for the latest information about the ERC32.

In paragraph 3 of this document system design possibilities are discussed along with more detailed descriptions of system components. Different system block configurations are presented and the interfaces between system components are examined.

In paragraph 4 the demonstration board DEM32 built around the ERC32 is presented. The detailed design is presented along with a functional description of the board.

Note that this document describes the ERC32 system using rev. A of the MEC.

2. DOCUMENTS

[ERCSPC] Specification for a 32-bit embedded computing core (ERC32),
WDI/JG/1334/NL, Issue 3, ESA/ESTEC, 29-05-1991.

[IUSPC] IU-RT Preliminary Device Specification, AMS/IURT/0017/CLG, issue 6,
TEMIC / MHS, August 28th 1995.

[IUUM] TSC691E Integer Unit User´s Manual, Rev. H,
TEMIC / MHS, December 2 1995.

[FPUSPC] FPU-RT Preliminary Device Specification, AMS/FPURT/0018/VS, issue 6,
TEMIC / MHS, August 28th 1995.

[FPUUM] TSC692E Floating Point Unit User´s Manual, Rev. H,
TEMIC / MHS, December 2 1995.

[MECSPC] MEC Device Specification, MCD/SPC/0009/SE, issue 4,
SAAB Ericsson Space, April 10 1997.

[DEM32/3] DEM32 V.3 USER'S MANUAL, MCD/TNT/0021/SE, issue 3,
SAAB Ericsson Space, Januari 28 1997.

[ATAC1] ATAC 2.0 Ada Tasking Coprocessor Data Sheet, draft issue 0.99,
March 1993.

[ATAC2] ATAC 2.0 Ada Mechanism Performance, draft issue1.0, March 1993.

[TAPSPC] Standard Test Access Port and Boundary-Scan Architecture, IEEE-STD-1149.1, June 20 1989

[ERCSYS] ERC32 System Design Document, MCD/TNT/0009/SE,
SAAB Ericsson Space, 20 Apr 1994

[ERCFNC] ERC32 Functional Description, MCD/TNT/0005/SE,
SAAB Ericsson Space, 20 Apr 1994

[ERRDET] Concurrent Error-detection for SPARC, WDI/JG/1734/NL, Issue 1,
ESA/ESTEC, 25-11-1992.

[AMON1] AdaWorld Development Environment for SPARC based Workstations
under Solaris 2 to ERC32 Targets, release 5.5.4,
Cross-Development Guide, MCD-ALS-P2-DOC-015, issue 2 revision 1,
Thomson Software Products / Alsys, December 12 1996.

[GMON1] The ERC32 GNU Cross-Compiler System, Ver. 1.0,
ESA/ESTEC, October 1996

(This page is intensionally left blank.)

3. SYSTEM DESIGN CONSIDERATIONS

3.1. Introduction

A basic computer implementation using the ERC32 chip set would consist of the following components:

* ERC32 core, including:

- Processor, which consists of one Integer Unit (IU) and one Floating Point Unit (FPU). The processor includes concurrent error detection facilities.

- Memory Controller (MEC), which is a unit consisting of all necessary support functions such as memory control and protection, EDAC, wait state generator, timers, interrupt handler, watch dog, UARTs, and test and debug support. The unit also includes concurrent error detection facilities.

- One or, if a separate watchdog clock source is required, two oscillators.

- Buffers and latches necessary to interface with memory.

* Memory

* Serial communications (UART) interface circuitry

The above configuration is the simplest possible and is probably only usable for performance measurements etc. A computer in a space application is most likely to have some additional Input/Output interfaces apart from the serial communication implemented by the UART function of the MEC.

In addition to the basic bus interface, an exchange memory area providing simultaneous access from the ERC32 and the external bus is supported by ERC32 and could thus be easily implemented.

For efficient Ada code execution, an Ada Tasking Coprocessor (ATAC) might also be added to the system.

All of the above functions typically fits onto one printed circuit board in space applications, depending on memory size. For high-reliability applications, the concurrent error checking capability of ERC32 could be used. This would then double the no. of IU/FPU components in the system.

In paragraph 3.3 various ERC32 computer configurations are presented.

3.2. ERC32 Functions Overview

The IU-RT / FPU-RT provides SPARC V7 compatible processing functions with extensions for error-detection. The MEC supplies the processor support functions.

- System start up control and reset
- Power down mode control
- System clock
- Watchdog function
- Memory interfaceto RAM ranging from 256 kbyte to 32 Mbyte
- Memory interface to PROM ranging from 128kbyte to 16 Mbyte
- I/O interface to exchange memory (e.g. DPRAM) ranging from 4 kbyte to 512 kbyte.
- I/O interface to four peripherals
- DMA interface
- Bus arbiter
- Programmable wait-state generator
- Programmable memory data write access protection including support for block protection.
- Memory redundancy control
- EDAC, with byte and halfword write support
- Trap handler including 15-level interrupt controller
- One 32-bit general purpose timer with 16-bit scaler
- One 32-bit timer with 8-bit scaler (Real-Time-Clock)
- UART function with two serial channels
- Built-in concurrent error detection including support for master checking of IU and FPU
- System error handler
- Parity control on system bus
- Test and debug support.

3.3. ERC32 System Configurations

3.3.1. The Basic ERC32-based Computer

The first thing to consider when selecting the ERC32 core configuration is the system requirements on memory. The MEC provides for a wide variety of RAM organisations, supporting device memory sizes ranging from 8kx1 bits to 8Mx8 bits. Parity and EDAC protection of RAM is programmable in the MEC. In most systems EDAC protection of RAM is desired.

The non-volatile memory used for system boot can be implemented as a normal word-wide EDAC protected memory. The PROM data bus then has a width of 32 bits accompanied by 8 check bits. This configuration, shown in Figure 1~~1~~ below, yields maximum performance for Boot PROM access. Note that the Boot PROM and RAM share the same buffers.

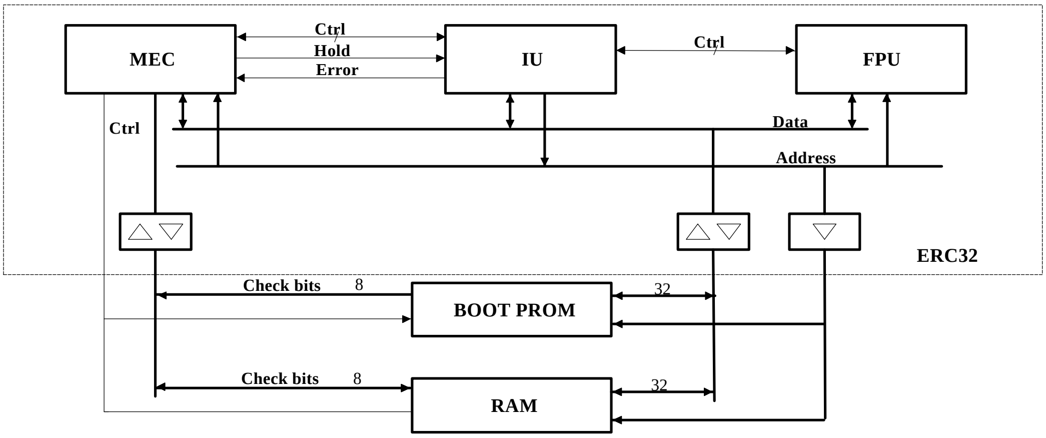


Figure 1~~1~~ - Basic Configuration with 40-bit Boot PROM

In many systems board space, power consumption and component cost are driving factors. The 40-bit wide Boot PROM typically requires 5 devices. The MEC provides an option to minimise the number of Boot PROM devices by accessing the PROM with a bus width of 8 bits. In this configuration, the MEC will perform four accesses to form one 32-bit word and feed it to the IU/FPU. The drawbacks of this option, shown in Figure 2~~2~~ below, are that the performance is divided by four for Boot PROM access and that no check bit protection is provided.

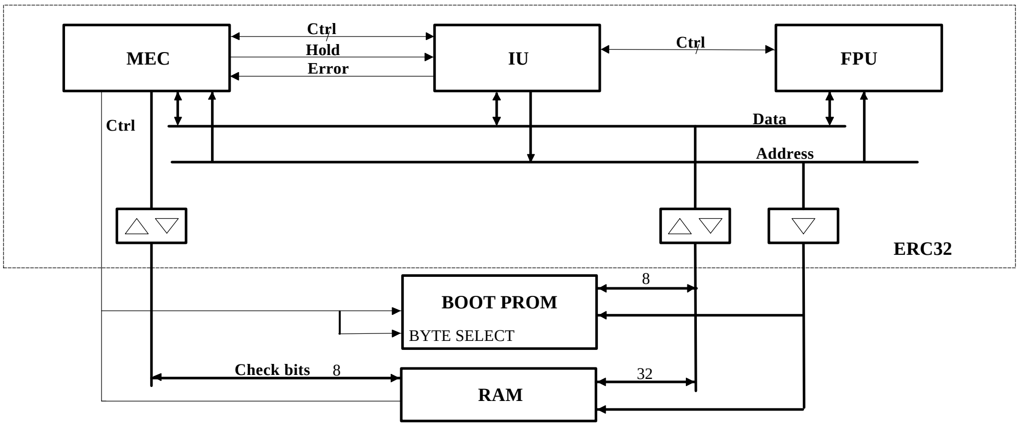


Figure 2~~2~~ - Basic Configuration with 8-bit Boot PROM

The MEC supports programming of boot PROM if implemented with EEPROM technology, both for manufacturing pre-programming and embedded system maintenance reprogramming.

3.3.1.1. ERC32 Parity

The ERC32 chipset is designed to allow for many system configurations. The design information provided in [ERCFNC] and the IU/FPU/MEC specifications is not focused on typical applications. The following is a discussion on the different parity checking options the system, described in Figure2~~2~~, can have.

3.3.1.1.1. IU to FPU Parity Check

Apart from the signals connecting the IU and FPU in a standard version 7 SPARC system, ERC32 provides an additional set of signals for error detection. If the 601MODE/602MODE signals are not asserted by the IU/FPU, parity will be checked on the control signals between the IU and FPU. The parity is provided on FIPAR/IFPAR signals. The HWERROR signal serves as a status output to indicate parity errors. Forcing 601MODE/602MODE signals low will disable the parity checking of all input signals. The IU and FPU will operate with the standard input signals. The generation and checking of internal parity is still active.

3.3.1.1.2. IU/FPU to MEC Parity Check

If parity check is enabled by the MEC (if the NOPAR* signal is deasserted and RAM memory parity protection is enabled by programming the MEC) and the 601MODE/602MODE signals are not asserted by the IU/FPU, parity will be checked on the following signals: the address bus (APAR), the data bus (DPARIO), the control signals ASI and SIZE (ASPAR), and the control signals LDSTO, DXFER, LOCK, WRT, RD, and WE* using the IMPAR signal. Note especially that the IU/FPU signal DPAR shall be connected to the MEC signal DPARIO.
Assertion of the NOPAR* signal will force the MEC to disable the parity checking of all signals related to the ERC32 local buses.

3.3.1.1.3. Memory and I/O to MEC Parity Check

If parity is enabled for memory and I/O (programmable in the MEC), the parity bit of the data word is emitted on the MEC signal DPARIO during store operations. During read operations, the parity bit of the data word is expected on the same signal DPARIO.

Read access with no parity
In the case that no parity is supplied, the MEC calculates the parity and drives the DPARIO pin towards the IU/FPU, see Figure3~~3~~.

Write access with no parity
In this case, the IU/FPU will drive the DPARIO pin on the MEC. The MEC will check parity on the data driven by the IU/FPU. No parity signal is used towards memory and I/O, see Figure 3~~3~~.

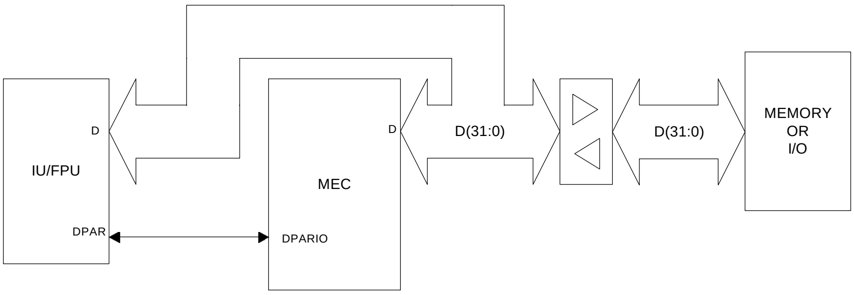


Figure 3~~3~~ - Memory and I/O access with no parity

Read access with parity
In the case that parity (or parity and EDAC) is used, the MEC will reflect the state of the DPAR on DPARIO towards the IU/FPU, see Figure4~~4~~.

Write access with parity
In this case, the IU/FPU will drive the DPARIO pin on the MEC. The MEC will check parity on the data driven by the IU/FPU. The DPAR signal is reflected on the DPARIO pin, see Figure4~~4~~.

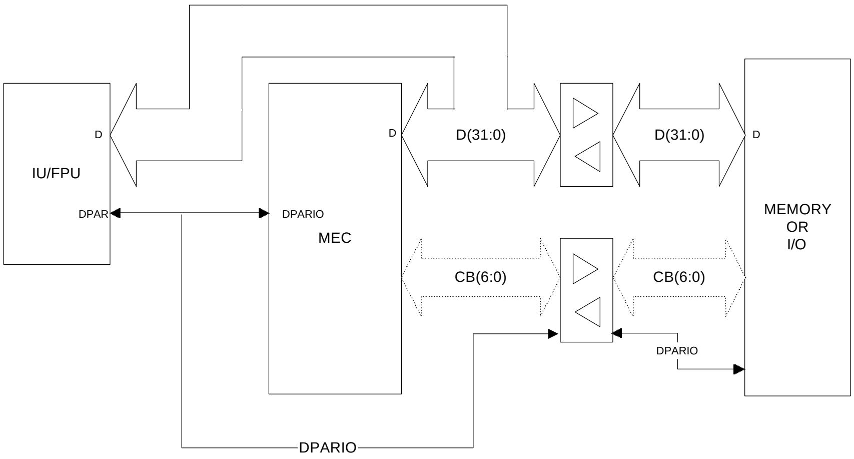


Figure 4~~4~~ - Memory and I/O access with parity

3.3.1.1.4. Memory to MEC Parity Implementation

If parity is enabled for memory (programmable in the MEC), the parity bit of the data word is emitted on the MEC signal DPARIO during store operations. Since there are seven EDAC bits, the straight forward approach would be to store the memory parity bit in the same memory device as the EDAC bits. However, due to the fact that the timing of EDAC checkbits and DPARIO is not the same, some glue logic is needed or the DPARIO has to be stored in a separate memory device.

There are a number of possible system configurations and associated problems:

1. *For systems with no PARITY and EDAC protection on memory*:
   No memory device is needed for storage of EDAC/DPARIO. Systems with no EDAC/PARITY protection on memory does not have a timing problem.

2. *For systems with PARITY but without EDAC protection on memory*
   Use separate memory device (one bit memory) for the DPARIO or
   use 9-bit memory component for combined Data(0..8) + DPARIO storage.
   DPARIO is stored in the separate memory device using the MEMWR1* signal.

3. *For systems with PARITY and EDAC protection on memory*:

   a. *Systems with separate memory device for DPARIO storage.*
      DPARIO is stored in the separate memory device using the MEMWR1* signal.

   b. *Systems with memory device for combined storage of CB(0..6) + DPARIO.*
      Use a transceiver/register to latch DPARIO using the MEMWR1* signal before storing to memory using the MEMWR2* signal.

   c. *Systems with memory device for combined storage of CB(0..6) + DPARIO.*
      Use one waitstate to store the DPARIO signal to memory using the MEMWR1* signal only.

The DEM32 design is using the 3c solution and the timing analysis in chapter 4.3.~~4.3.~~ is done based on this solution.

3.3.2. ERC32 Computer with ATAC and General Purpose I/O

Figure 5 shows a tentative system with I/O peripherals and an Ada Tasking Accelerating Coprocessor (which is also an I/O unit from a system point of view). Note that separate Address and Data buffers probably are required for I/O and memory. The MEC signals IOBEN and MEMBEN are used for this purpose. If common buffers should be used, IOBEN and MEMBEN would have to be gated, resulting in a timing performance loss.

There are no special I/O instructions in the SPARC architecture. Therefore, a memory mapped I/O bus is used. Also, since the memory addressing range with 32address bits is more than required in the foreseeable future for space projects, part of the memory address range is reserved for I/O accesses.

Memory external to the ERC32 core can be implemented as I/O units allowing longer access times in case they are connected to an external bus, e.g. a VME-bus. A reason for connecting external memory in this way is that the chip select signals for the internal memory are specially defined and decodes no more than 32Mbytes of memory.

The I/O communication may require widely different access times and therefore a ready signal (BUSRDY*), for signalling that the transfer has been accepted, is provided. The ready signal allows the processor to continue the processing as soon as the ready signal is received without unnecessary waiting. To avoid the disadvantage that the I/O units have to provide such a ready signal even in case their timing is fixed, a combination of wait-state generator and ready signal can be used. The wait-state generator is programmed to give the minimum time to wait. Then, if an I/O unit needs more time it can use a bus ready (BUSRDY*) signal indicating that the processor has to wait further. For I/O units not requiring extra time this signal should be asserted immediately upon access, indicating bus ready. The MEC can be programmed to insert from one to fourteen waitstates before the BUSRDY* signal is sensed. An I/O unit which may occasionally require more time, has to deassert the BUSRDY* signal within the programmed access time to make the processor wait.

A Bus Time-out function is included in the MEC in order to eliminate the risk for blocking the system if an I/O unit permanently deasserts the BUSRDY* signal.

A typical I/O unit, including the ATAC, will have interrupt capability towards the IU. In the MEC a maximum of 5 external interrupts are provided. In a system including an ATAC, additional interrupts are provided to be handled by the ATAC task manager. These interrupts will however have a greater latency than the interrupts directly connected to the MEC.

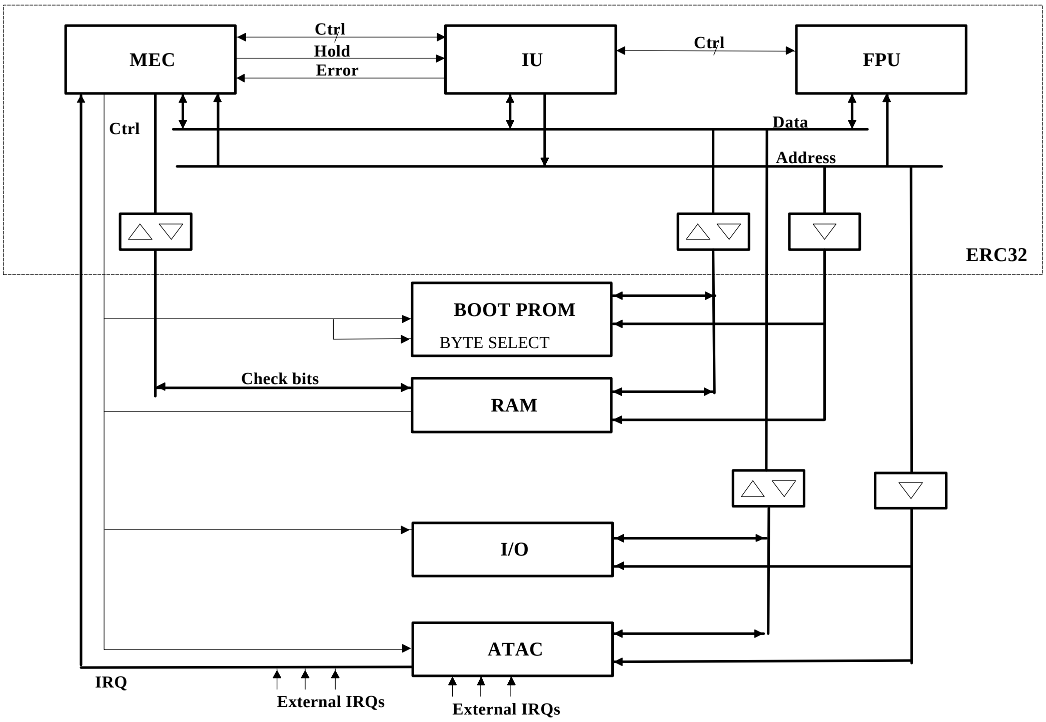


Figure 5~~5~~ - System with I/O and ATAC

Complete information about the Ada Tasking Accelerating Coprocessor (ATAC) can be found in the ATAC guides [RD10-12]. The ATAC is connected as an I/O unit but some glue logic is required to adapt to the MEC signal timing. In particular, the bus cycle length varies when accessing the ATAC. Therefore, the number of wait-states is set to a minimum in the MEC, but the RELEASE signal from the ATAC must be used to create the BUSRDY* signal with proper timing towards the MEC to extend the bus cycle as required. Note that in the I/O-area the MEC will always insert one waitstate to wait for the BUSRDY* signal, even when the number of waitstates is set to the minimum.

The ATAC does not generate or check the parity on the address and data lines. Therefore, the MEC must be programmed to supply a parity bit for ATAC accesses.

The ATAC has two interrupt request outputs, ATTN and FAULT. These signals can be connected to external interrupt inputs on the MEC. Note that these signals are active high while the default polarity for external interrupts in the MEC is active low, i.e. the MEC must be programmed to be active high for the external interrupts used for ATTN and FAULT. Moreover, the ATAC removes the asserted ATTN signal only when it receives an acknowledge signal. The external interrupt acknowledge output from the MEC (EXTINTACK) is to be used for this purpose. Note that the MEC must be programmed for this purpose and also it must be programmed to use edge detection for ATTN.

FAULT indicates faults detected internally in the ATAC and should be connected to a non-maskable or an unmasked interrupt input on the MEC.

3.3.3. ERC32 Computer with Exchange Memory

In ERC32 systems with multiple units that can access the main memory, outside the ERC32 core control there might be indeterminism. As an example an ERC32 core could be connected to a asynchronous system bus, where the masters of the system bus can access the local RAM of the ERC32 core through DMA. This architecture leads to a number of problem areas.

For instance; as the arbitration of the system bus is handled outside the ERC32 core simultaneous accesses might cause a deadlock in the arbiter. This could be solved by aborting either the local request or the system bus slave request, which however is complicated and generally requires software overhead.

A better solution would be to implement a separate area with a separate arbiter for exchange of data. The MEC supports this concept by dedicating a memory area called the Exchange Memory area.

An exchange memory can be implemented and used in various ways. The most common implementation is as an non-intrusive, high performance, data exchange channel. The MEC therefore treats the exchange memory as a true dual port memory with 32-bit organisation. Thus, the exchange memory should only be used for transferring data, i.e. programs should not be executed from the exchange memory.

When it is not possible for an application to use true dual port RAM to implement the exchange memory, a "true" dual port memory must be emulated outside the MEC, i.e. the MEC does not support any special arbitration for logical dual port RAM.

The only difference between exchange memory accesses and main memory accesses is that the MEC has to wait for the "BUSY" signal from the DPRAM, connected to BUSRDY* of MEC, before the write strobes can be activated. For this reason the MEC will introduce address wait-states (i.e. wait for BUSY) when accessing the exchange memory. After the address wait-states the MEC knows whether the access is in conflict with another unit or not (as indicated by the BUSY signal).

This means that in the exchange area the function of the BUSRDY* signal is such that when BUSRDY* is deasserted the exchange memory is considered busy. The delay of the BUSY signal depends on the dual port RAM chosen.

A typical dual port RAM needs less than 80 ns to produce the BUSY signal and therefore it is typically sufficient that the MEC waits one clock cycle at the start of the access for the assertion of BUSRDY* signal. The address wait-state phase will of course slow down the accesses towards the exchange memory, but on the other hand the address decoding and protection mechanism can run in parallel and therefore it might be possible to perform the later part of the access with zero wait-states. If BUSRDY* is asserted in the beginning of the second cycle, the normal wait-state controlled access continues. If BUSRDY* is deasserted during the address wait-state time, the MEC will delay the access until the BUSRDY* signal has been asserted and continue with the programmed normal data wait-state controlled access.

The minimum length of an exchange memory access is three clock cycles.

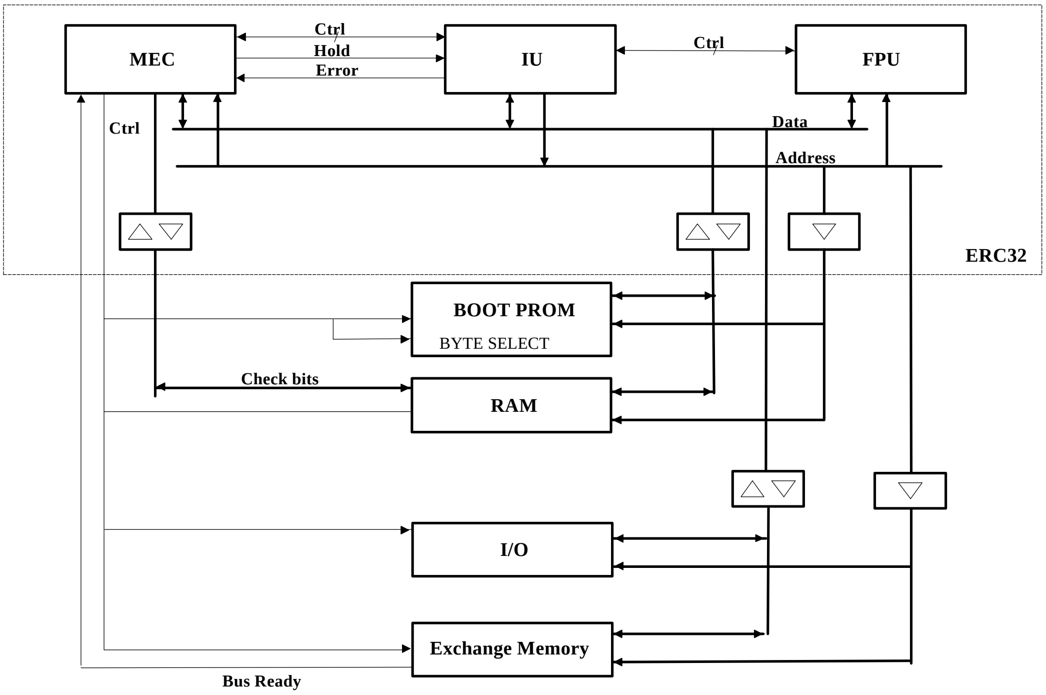


Figure 6~~6~~ - System with I/O and Exchange Memory

3.3.4. ERC32 Computer with DMA Slave Interface

Figure 7~~7~~ shows a tentative system with a slave DMA interface.

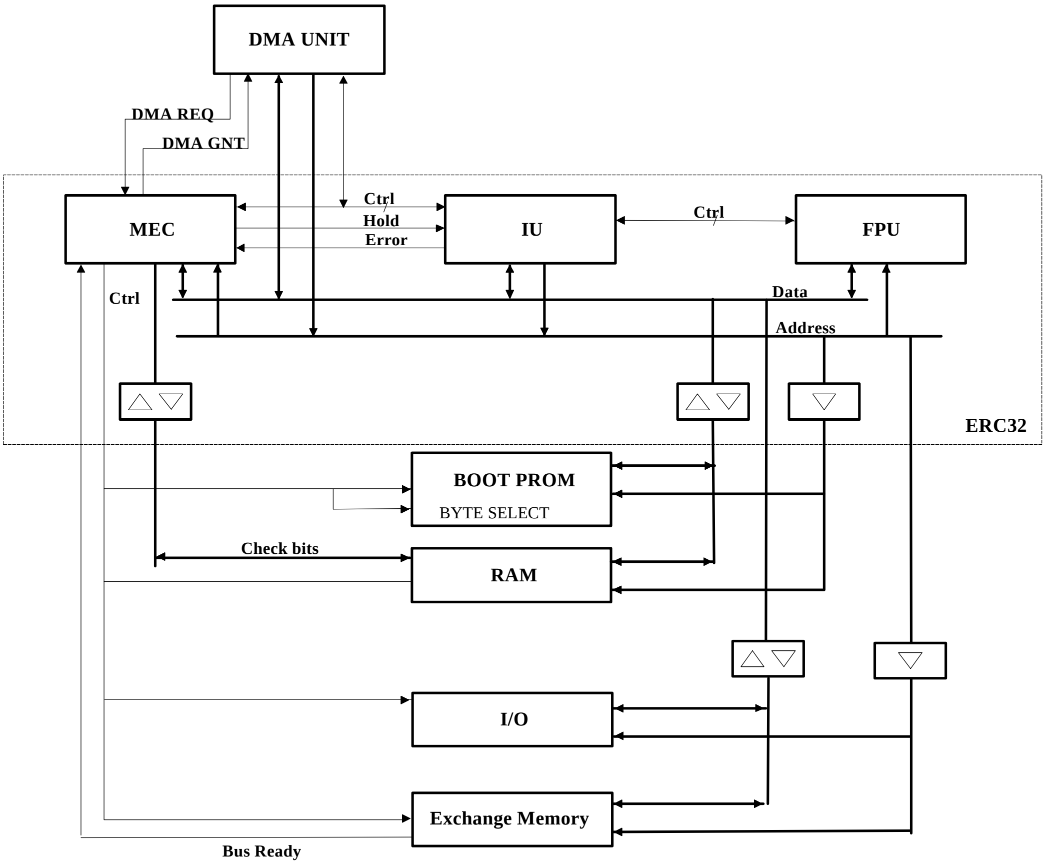


Figure 7~~7~~ - System with Slave DMA Interface

The DMA unit must, once access has been granted, drive all memory access control signals that the IU normally drives.

In case of time criticality during execution of a special region, the application program might need to run without disturbance from a DMA access.

The concurrent error detection on system level is able to detect faults in DMA accesses, either the DMA itself flags an error or the MEC can detect erroneous behaviour such as if the DMA does not remove the DMAREQ*.

It is possible to transfer up to 341 words within 1024 system clock cycles with a DMA. If one extra WS is needed for the DMA it is possible to transfer up to 256 words within 1024 system clock cycles. Note also that only the DMA will sample the Memory Exception output during DMA accesses.

If the DMA occupies the system bus for a longer time than expected, the system can be alerted that something unexpected has occurred.

The time-out is needed to avoid deadlocks, or unexpected long access time on the bus.

Such extended access time lead to indeterministic real time behaviour of the system, and must therefore be aborted with a synchronous trap. In case of IU access, the IU both aborts the cycle and handles the trap. In case of a DMA, the DMA is responsible for aborting the bus cycle, but can not process the trap. Thus the trap issues an interrupt to the IU via the MEC.

For the ERC32 system detection of erroneous accesses is essential to guarantee the integrity of the processes.

4. The DEM32 Board

4.1. Introduction

This chapter describes the DEM32 evaluation board developed within the 32 Bit Microprocessor project. DEM32 is a computer board with the purpose to demonstrate the radiation tolerant SPARC V7 compatible chip set of IU-RT, FPU-RT and MEC.

4.2. DEM32 Functional Description

The DEM32 is a computer board based on the ERC32 processor core. The purpose of the board is to demonstrate the ERC32 core capabilities and it is also intended to be used to develop application software. The DEM32 is an industry-standard double Europe size (160 mm x 233,4 mm) printed circuit board. The board may be inserted into a standard VME crate for power supply, but power may also be supplied with a custom connector.

The DEM32 has the following characteristics:

- ERC32 computer core: IU, FPU and MEC
- 2 Mbyte RAM
- 512 kbyte start-up Flash EEPROM (also called boot PROM)
- Ada Tasking Accelerating Coprocessor (ATAC) [optional]
- Two RS232C standard serial ports

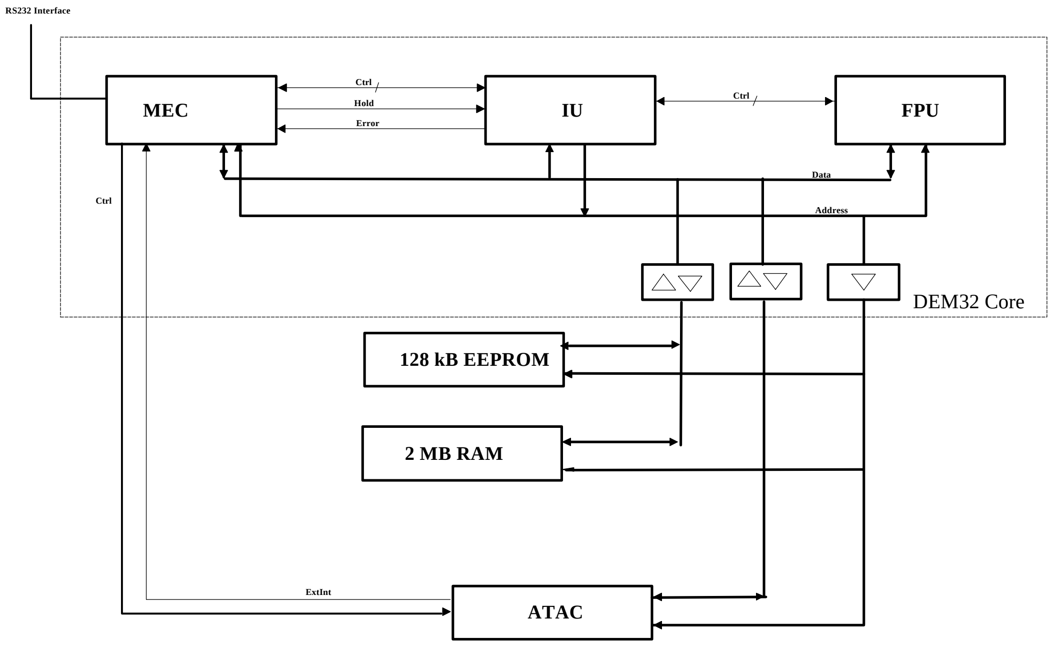


Figure 8~~8~~ - DEM32 Block Diagram

The full schematics for the DEM32 are presented in Appendix 2. The DEM32 includes one PLD of type 22V10 named ATACIF implementing the handshake glue logic to the ATAC. The contents of the PLD is listed in Appendix 1 (ABEL^{TM} source code).

4.2.1. ERC32 Core

The block diagram of the ERC32 core architecture is shown in Figur9~~9~~.

- Integer Unit (IU-RT)
- Floating Point Unit(FPU-RT)
- Address latches and data buffers
- MEmory Controller (MEC)

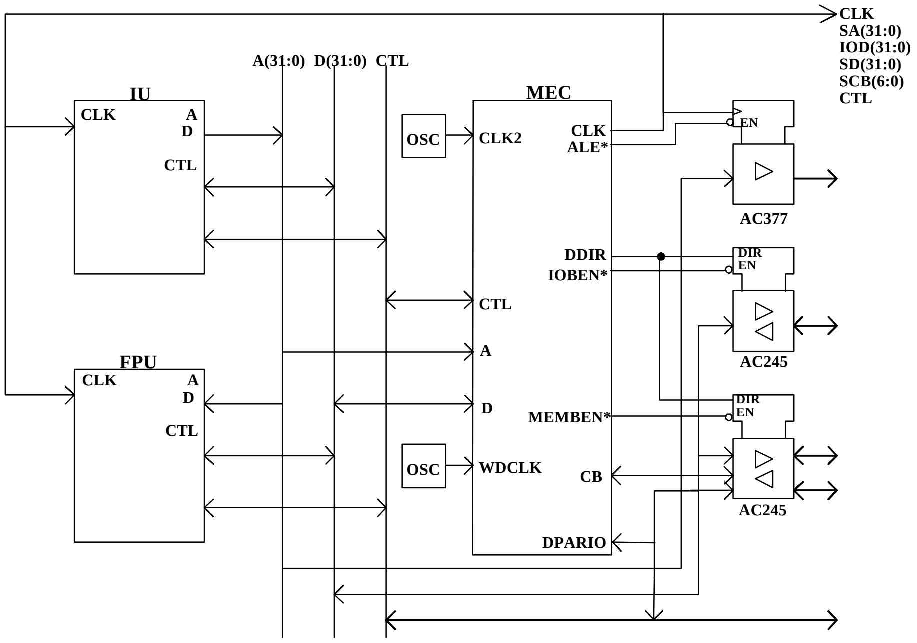


Figure 9~~9~~ - ERC32 core

4.2.1.1. Integer Unit (IU-RT)

The IU is the primary processing engine in the SPARC architecture, executing all instructions except for specific floating-point operations. The FPU performs floating-point calculations concurrently with the IU. The architecture also allows for concurrent operation through the use of an optional second coprocessor (not implemented in the DEM32). For a detailed description of the IU, see [IUSPC] and the IU User´s Manual [IUUM].

Significant features of the IU include:

- Full binary compatibility with entire SPARC V7 application software base
- Architectural efficiency that sustains 1.25 to 1.5 clocks per instruction
- Large windowed register file
- Tightly coupled floating-point interface
- User/supervisor modes for multitasking
- Semaphore instructions and alternate address spaces for multiprocessing
- Tagged arithmetic instructions to support artificial intelligence software

The IU supports concurrent error detection by including parity generation and checking, program flow control, and possibility to work in a master/slave configuration. In the DEM32 it is possible to enable program flow control by forcing this input low at the board test connector.

4.2.1.2. Floating Point Unit (FPU-RT)

The FPU provides high-performance, IEEE STD-754-1985 compatible single- and double-precision floating-point calculations for ERC32 systems, and is designed to operate concurrently with the IU. All address and control signals for memory accesses by the FPU are supplied by the IU. Floating-point instructions are addressed by the IU, and are simultaneously latched from the data bus by both the IU and FPU. Floating-point instructions are concurrently decoded by the IU and the FPU, but do not begin execution in the FPU until after the instruction is enabled by a signal from the IU. Pending and currently executing FP instructions are placed in an on-chip queue while the IU continues to execute non-floating-point instructions.

The FPU has a 32 x 32-bit data register file for floating-point operations. The contents of these registers are transferred to and from external memory under control of the floating-point load/store instructions. Addresses and control signals for data accesses during a floating-point load or store are supplied by the IU, while the FPU supplies or receives data. Although the FPU operates concurrently with the IU, a program containing floating-point computations generates results as if the instructions were being executed sequentially.

The FPU supports concurrent error detection by including parity generation and checking, and possibility to work in a master/slave configuration.

For more details on the FPU refer to the FPU device specification [FPUSPC] and the FPU User´s Manual [FPUUM].

4.2.1.3. Memory Controller (MEC)

The Memory Controller (MEC) interfaces directly with the address, data, and control buses of the IU and the FPU. It generates the system clock, reset and control signals for the processor elements, and all strobes for the memory and the Input/Output ports. Also, it recognises external interrupt inputs and includes two serial ports, UARTs. For a detailed description of the MEC refer to the MEC device specification [MECSPC].

4.2.1.3.1. External Bus Interface

The on-chip bus controller generates all necessary strobes required by external memory, IU and FPU, and external I/O units (ATAC).

4.2.1.3.2. Trap Handling

The main objective for the ERC32 trap handling is to resolve internal hardware errors and memory access errors. Memory access errors are signalled by assertion of the MEXC* signal which force the IU to vector a data access exception. For the memory access errors, the MEC FAR (Failing Address Register) is the address of the last synchronous data error detected for IU or DMA.

The MEC is capable of handling nine different fault events which all result in a synchronous trap (assertion of MEXC*).

1. Parity error on control bus
2. Parity error on data bus
3. Parity error on address bus
4. Access to protected area
5. Access to unimplemented area
6. MEC register access violation
7. Uncorrectable and detected error in memory
8. Bus time-out
9. System bus error

The MEC handles fifteen different events which all represent asynchronous trap (assertion of the Interrupt Request Level IRL inputs of the IU).

1. Watch Dog time-out
2. DMA time-out
3. DMA access error
4. UART error
5. UART A data ready
6. UART B data ready
7. RTC interrupt
8. General Purpose Timer interrupt
9. Correctable error in memory
10. Masked hardware error
11. Five external interrupts

4.2.1.3.3. Timers

The MEC includes a timer clock divider, that supplies the timer functions with a timer clock derived from the system clock. The timers have configurable scalers and counters.

The MEC has two timers:
- One real-time clock timer.
- One general purpose timers.

4.2.1.3.4. Serial Channels

The MEC supplies the board with two asynchronous serial communication channels. These channels implement full duplex serial communication. The serial communication channel baud rate is programmable.

4.2.1.3.5. Watchdog Timer

The watch-dog function is realised as a retriggerable single shot timer with programmable time-out. A reset timer starts counting when the watch-dog timer has elapsed. The reset triggering can be avoided if the watch-dog interrupt is refreshed before the reset timer elapses. There is also a "trap door" function which disables the watchdog.

4.2.1.3.6. Power Down Mode

The MEC includes a software programmable power down mode that puts the system in a power saving mode by inactivating of the IU/FPU bus controls. The system wakes up from Power down mode if any enabled interrupt is detected.

4.2.1.3.7. I/O Ports

Four general purpose I/O ports are supported by the MEC. Each port can be individually controlled by programming an internal MEC register. On the DEM32 only one port is implemented, I/O port 0 which is used for the ATAC interface.

4.2.2. Buffers

The address bus from the IU is latched and buffered using D flip-flops with clock enable (4 x 54AC377). The memory data bus (RAM and EEPROM) including check bits is buffered (5 x 54AC245). The I/O data bus (ATAC) has separate buffers (4 x 74AC244).

4.2.3. Clock Oscillators

The MEC provides the IU and FPU with a system clock. This clock is derived from the external system oscillator. The oscillator operates at a frequency of 20 MHz, which means that the system clock frequency is 10 MHz.

The watchdog oscillator is connected to the WDCLK input of the MEC. The watchdog oscillator frequency is 1,8432 MHz.

4.2.4. Ada Tasking Accelerator Coprocessor (ATAC)

In Figure 10~~10~~ below the block schematics of the DEM32 ATAC interface is shown. The ATAC cannot be connected to the ERC32 system bus directly, handshaking logic must be implemented. In the DEM32 the handshaking is implemented in a PLD, 22V10 (see Appendix 1).

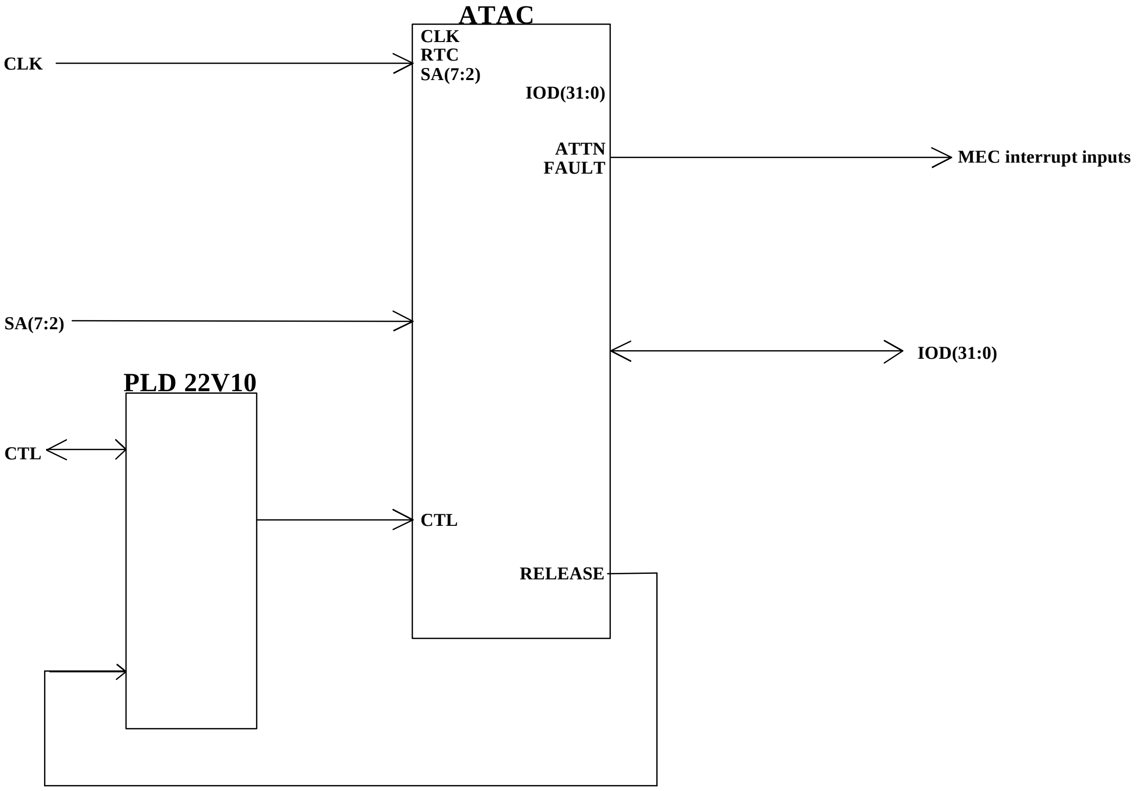


Figure 10~~10~~ - DEM32 ATAC Block

The ATAC is addressed as I/O unit 0 in the DEM32, i.e. command address 0 of the ATAC is accessed at the first memory address of I/O unit 0. The chip is operated at 10 MHz which is the system clock frequency. The RTC input of the ATAC is connected to the WDCLK signal. ATAC external memory is not supported on the DEM32. The EXTCLK input of the ATAC is not used, nor are the external interrupt inputs of the ATAC. For details about the ATAC see [ATAC1-3]. The ATTN and FAULT interrupts from the ATAC are connected to MEC external interrupt inputs, and INTACK is connected to the EXTINTACK output of the MEC.

4.2.5. Memory

In Figure 11~~11~~ below the block schematics of the DEM32 memory is shown.

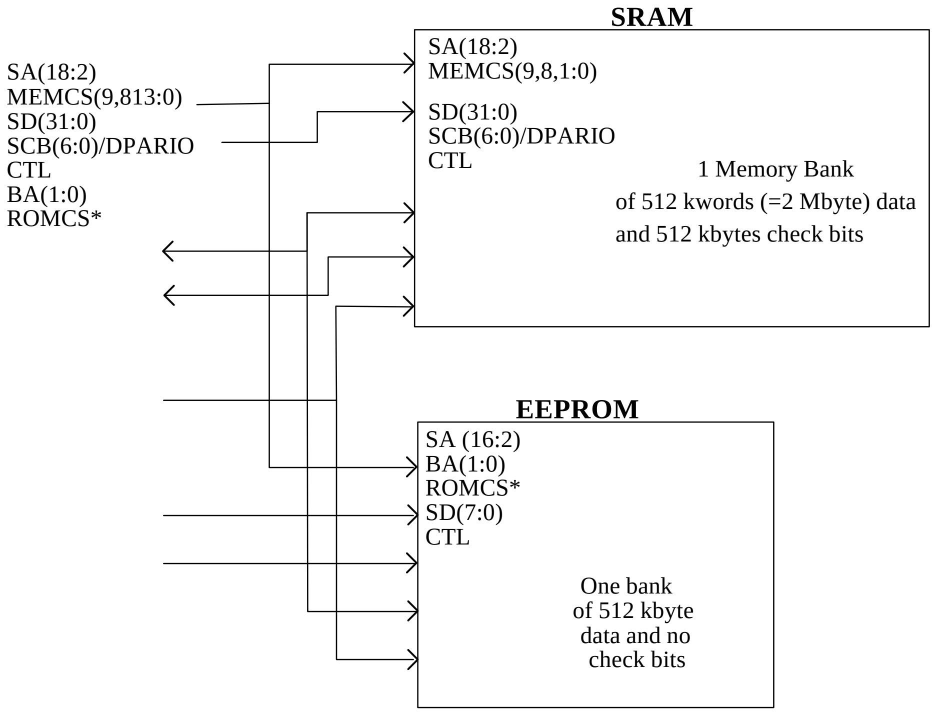


Figure 11~~11~~ - DEM32 Memory

4.2.5.1. Nominal RAM

The board is equipped with 2 Mbytes of nominal RAM (excluding checkbits). The RAM is implemented as one bank with four 512k x 8 utilising MEMCS*(0) signal of the MEC. Once the Memory Configuration Register of the MEC has been initialised it is seen as a continuous 2 MBytes segment by software. The RAM may be accessed at zero wait states (programmable in the MEC).

4.2.5.2. Memory Expansion

In addition to the 2 Mbytes of nominal RAM, an expanded RAM area is reserved in the board. The DEM32 is able to access totally 4 Mbytes, two separate nominal RAM banks of 2 Mbytes utilising MEMCS*(0) and MEMCS*(1) signals and one redundant bank of 2 Mbytes utilising MEMCS*(8) signal (excluding checkbits). As mentioned before, only one nominal RAM bank of 512 kwords (= 2 Mbyte) and 512 kbytes check bits are implemented.

4.2.5.3. Start-Up PROM

The board is equipped with 512 kBytes of start-up PROM, byte wide organised (8-bit option set in MEC). The PROM is realised with one 512k x 8 Flash EEPROM allowing access with one wait state.

The start-up PROM of the DEM32 contains system start up tests and the monitor [AMON] or [GMON] which has been adapted to the DEM32 board.

4.2.6. DEM32 Memory Map

The memory map of the DEM32 is defined by the MEC. Only a small part of the possible MEC memory configuration is implemented in the DEM32. The memory map of the DEM32 is programmed on start-up as shown in Figure12~~12~~.

Address (hexadecimal)	Memory contents	Size (Bytes)	Data size and parity options
0x00000000	Boot PROM	512k	8-bit mode 8 to 32 bit conversion No parity Only byte write
0x00080000	Non-implemented Boot PROM area	15,5M	
0x01000000	Non-implemented Extended PROM area	15,5M	
0x01F00000	Non-implemented Exchange memory	512k	
0x01F80000	MEC Registers	512k	Parity only Word write and Word/ Hword/ Byte read
0x02000000	RAM Memory	2M	Parity/EDAC options All data sizes allowed
0x02200000	RAM Expansion Memory [Option]	2M	Parity/EDAC options All data sizes allowed
0x02400000	Non-implemented RAM area	28M	
0x04000000	Non-implemented Extended RAM area	192M	
0x10000000	I/O area 0, ATAC [Option]	512	Parity option All data sizes allowed
0x10000200	Non-implemented I/O area 0	(16M-512)	
0x11000000	Non-implemented I/O area 1	16M	
0x12000000	Non-implemented I/O area 2	16M	
0x13000000	Non-implemented I/O area 3	16M	
0x14000000	Non-implemented Extended I/O area	1728M	
0x80000000	Non-implemented Extended general area	2G	

Figure 12~~12~~ - DEM32 Memory Map

4.2.7. Resident Software in Start-Up PROM

Upon system reset of the DEM32, performed on power-on or by pushing the RESET button on the front panel, execution is started in the Start-Up Prom. The Start-Up PROM contains a small boot-strap routine, hardware initialisation, system self-test and debugger/monitor, see [AMON] or [GMON].

4.2.7.1. AlsyMon Boot Strap

A small boot strap routine is executed immediately after system reset. The boot strap routine initialises some MEC registers and copies the contents of the PROM to RAM, whereafter execution continues in RAM.

The following parameters are changed in the MEC register reset values:

MEC Control Register:

- Access protection disabled
- UART baud-rate 19200
- UART parity disabled
- UART even parity

MEC Memory Configuration Register:

- 2 RAM memory banks
- 1 MB RAM
- EDAC and Parity used on RAM
- PROM write disabled

MEC Waitstate Configuration Register:

- 0 waitstates on RAM read
- 0 waitstates on RAM write
- 1 waitstate on PROM read
- 1 waitstate on I/O 0 (ATAC)

MEC Interrupt Mask Register

- Masked Hardware errors interrupt enabled
- UART A Tx/Rx Interrupt enabled
- UART B Tx/Rx Interrupt enabled
- Correctable error in memory interrupt enabled
- UART error interrupt enabled
- DMA access error interrupt enabled
- DMA time-out interrupt enabled
- General Purpose Timer Interrupt enabled
- Real Time Clock Interrupt enabled

MEC Watchdog Trap Door Set

- Watchdog disable

Refer to [AMON1] for a detailed description and users manual of AlsyMon.

4.2.7.2. GCC rdbmon Boot Strap

A small boot strap routine is executed immediately after system reset. The boot strap routine initialises some MEC registers and copies the contents of the PROM to RAM, whereafter execution continues in RAM.

The boot strap loader operates in the following steps:

- The register files of IU and FPU are washed to initialise register parity bits.
- The MEC control, waitstate and memory configuration registers are initialised.
- The top 32K of the RAM is washed to initiate the EDAC checksums.
- Part of the loader is moved to the top of RAM to speed up operation.
- The remaining RAM is washed and the monitor is decompressed and installed.
- The text part of the monitor is write protected, except for the lower 4K where the traptable resides.
- Finally, the monitor is started.

The following parameters are changed in the MEC register reset values:

- PROM size is set to 512 Kbyte.
- PROM wait state value is set to 2 .
- RAM size is set to 2 Mbyte
- RAM wait state value is set to 0
- RAM EDAC and parity protection is enabled
- Number of nominal RAM blocks are set to 1
- Watchdog counter is disabled
- The UART baud-rate is set to 19200

The GCC rdbmon has been adapted by ESTEC for the DEM32 board.
Refer to [GMON1] for a detailed description and users manual of rdbmon.

4.2.8. Front Panel Functions

Figure 17~~17~~ shows the front panel of the DEM32 board.

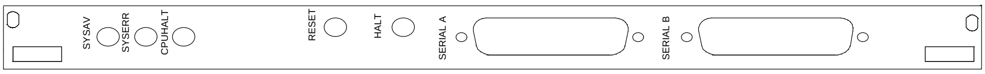


Figure 13~~13~~ - DEM32 front panel

Two RS232 type serial interface connectors, used for host computer communication, are provided on the front panel of the DEM32. One halt switch is provided. The halt switch shall be set to off (left) position for normal operation. A reset push-button used for hardware reset of the DEM32. Three LED indicators show the status of the DEM32. After power-up, the SYSAV LED should be lit and the SYSERR and CPUHALT indicators should be unlit.

4.2.8.1. Serial Interface

The electrical characteristics for the serial interface pins comply with the RS-232 standard, [RD9]. The Serial A connector corresponds to channel A of the MEC UART. Channel A is used by the monitor for host communication. The Serial B connector corresponds to channel B of the MEC UART. Channel B is used by the monitor for remote debugging. Note that no hardware handshake signals are provided. All handshaking must be implemented by software (e.g. XON/XOFF).

4.2.8.2. Reset Button

Pushing the reset button on the front panel will result in a hardware reset of the DEM32. A hardware reset is automatically performed when powering on the DEM32.

4.2.8.3. Halt Switch

There is a halt function in the DEM32. Setting the HALT switch in ON position (right) will halt execution in the DEM32 by asserting the SYSHALT signal of the MEC. During halt the IU, FPU, MEC including timers and UARTs are halted.

4.2.8.4. Indicators

There are three indicators on the front panel named SYSAV, SYSERR and CPUHALT. These indicators reflect the status of the corresponding pins of the MEC (see [RD7]).

SYSAV: System Availability. This LED is under software control. It is deasserted (unlit) by hardware reset. The monitor will assert SYSAV after successful completion of start-up tests. If the monitor detects any hardware error during start-up, SYSAV will remain unlit. SYSAV is automatically deasserted when CPUHALT is active.

SYSERR: System Error. This output is asserted whenever an unmasked error is detected by the MEC. This condition will normally lead to CPUHALT unless the MEC has been programmed to give reset. When running SPARCMon, CPUHALT is default. The error can originate either from the IU, FPU or the MEC.

CPUHALT: Halt Indication. This LED reflects the status of the HALT* inputs of the IU and the FPU. The CPUHALT can be generated either from the HALT button on the front panel, from software halt or from error halt (see SYSERR above).

4.2.9. DEM32 Board Layout

The DEM32 board layout can be found in [DEM32/3] DEM32 V.3 USER'S MANUAL, MCD/TNT/0021/SE, issue 3.

4.3. Timing Analysis

The timing analysis for the DEM32 has been performed for the three different access areas, RAM, EEPROM and I/O (ATAC). In appendix 2, the electrical schematics of the DEM32 board is shown. The signal names in the timing diagrams are as denoted in appendix 2. Unless otherwise noted, the worst case values have been used in the timing analysis. The timing for the IU is fetched from [IUUM], 14 MHz version. The timing for the MEC is fetched from the MEC Rev. A. All timing parameters named tx where *x* is a number refer to the MEC. For buffers and latches, military AC-logic timing has been applied. For the RAM, worst case values for MHS/M65656 (45 ns access time), which is a slower RAM than the RAM actually mounted on the board, has been used. The worst case values for MHS/M 65608 (25 ns access time) has been used as well.
For the EEPROM, worst case values for Hybrid Memory Product MEM8129 (150/200 ns access time) has been used.

Clock edges are denoted as follows:

$t_{a\ b\ c\ d}$ where

a = cycle no. as denoted in timing diagram referring to SYSCLK
b = SYSCLK or CLK2
c = ↑ denoting rising edge or ↓ denoting falling edge
d = edge no. within cycle for CLK2 (each SYSCLK cycle equals two CLK2 cycles)

The following timing parameters have been used in this analysis:

ac245:tp = Output Enable/Disable time for (54AC245, T = -55 to +125) = 10.5 ns
ac377:tpCkQ = Propagation Delay for (54AC377, T = -55 to +125) = 11.0 ns
ram:trGLQV = Output Enable Access Time =
= 20 ns for (M 65656, 45 ns) =
= 10 ns for (M 65608, 25 ns)
ram:trELQV = Chip-select Access Time =
= 45 ns for (M 65656, 45 ns) =
= 25 ns for (M 65608, 25 ns)
ram:trAVQV = Address Access Time =
= 45 ns for (M 65656, 45 ns) =
= 25 ns for (M 65608, 25 ns)
ram:trGHQZ = OE* high to high Z =
= 15 ns for (M 65656, 45 ns) =
= 8 ns for (M 65608, 25 ns)
ram:twDVWH = Data Setup Time =
= 25 ns for (M 65656, 45 ns) =
= 15 ns for (M 65608, 25 ns)
ram:twWHDX = Data Hold Time =
= 0 ns for (M 65656, 45 ns) =
= 0 ns for (M 65608, 25 ns)
mec:t9 = MEC internal MEMCS* valid Delay = 20 ns
mec:t10 = MEMCS* Output Delay = 15 ns
mec:t12 = MEMCS* Output latch Propagation Delay = 8 ns
mec:t13 = MEMBEN* Output Delay = 35 ns

mec:t15 = MEMWR* Output Delay = 35 ns
mec:t16 = OE* Output Delay = 35 ns
~~mec:t19 = Data setup time during store = 0 ns~~
mec:t19 = Data setup time (to SYSCLK-) during store = 20 ns
mec:t20 = Data hold time during store = 1 ns
mec:t21 = MEC Data Output Delay = 20 ns
mec:t22 = MEC Data Valid = 2 ns
mec:t23 = CB Output Delay = 35 ns
mec:t24 = CB output valid to high Z Delay = 2 ns
mec:t69 = SYSCLK Output Delay = 20 ns
mec:t71 = Data Setup Time MEC during Load = 3 ns
mec:t78 = Data Hold Time MEC during Load = 1 ns
iu:tDIS = Data Input Setup (TSC691E-14) = 7 ns
iu:tDIH = Data Input Hold (TSC691E-14) = 9 ns
iu:tDOD = iu:tpDClkB = Data Output Delay (TSC691E-14) = 35 ns
iu:tDOH = iu:tpDCkBI = Data Output Valid (TSC691E-14) = 4 ns

4.3.1. RAM Access

The RAM access timing is determined by the IU, MEC, buffer and memory timing. In Figure 14 the timing is shown for a load ~~and store~~ operation at 0 waitstates and store operation at 1 waitstates, 10 MHz system clock. For instance the address setup time for the address latches (AC377) is calculated in the diagram to 2.5 ns compared to t4 = 15 ns which is the minimum setup time required by the MEC.

4.3.1.1. Data Setup Time to IU and MEC During Load

Data shall be available at $t_{2SYSCLK\uparrow}$ with a setup time of mec:t71 = 3 ns.
Data shall be available at $t_{2SYSCLK\uparrow}$ with a setup time of iu:tDIS =7 ns.

Timing chain 1: MEMBEN* asserted to data available

Time from $t_{2CLK2\downarrow 1}$ to $t_{2SYSCLK\uparrow}$ = 75 ns + t69 = 75ns (worst case when t69 = 0)
Margin 1 (MEC) =
= ($t_{2SYSCLK\uparrow}$ - $t_{2CLK2\downarrow 1}$) - mec:t13 - ac245:tp - mec:t71 =
= 75 - 35 - 10.5 - 3 = 26.5 ns
Margin 2 (IU) =
= ($t_{2SYSCLK\uparrow}$ - $t_{2CLK2\downarrow 1}$) - mec:t13 - ac245:tp - iu:tDIS =
= 75 - 35 - 10.5 - 7 = 22.5 ns

Timing chain 2: OE* asserted to data available

Time from $t_{2CLK2\downarrow 1}$ to $t_{2SYSCLK\uparrow}$ = 75 ns + t69 = 75ns (worst case when t69 = 0)
Margin 3 (M 65656) (MEC) =
= ($t_{2SYSCLK\uparrow}$ - $t_{2CLK2\downarrow 1}$) - mec:t16 - ram:trGLQV - ac245:tp - mec:t71
= 75 - 35 - 20 - 10.5 - 3 = 6.5 ns
Margin 4 (M 65608) (MEC) =
= 75 - 35 - 10 - 10.5 - 3 = 16.5 ns
Margin 5 (M 65656) (IU) =

= ($t_{2SYSCLK\uparrow}$ - $t_{2CLK2\downarrow 1}$) - mec:t16 - ram:trGLQV - ac245:tp - iu:tDIS
= 75 - 35 - 20 - 10.5 - 7 = 2.5 ns

Margin 6 (M 65608) (IU) =
= 75 - 35 - 10 - 10.5 - 7 = 12.5 ns

Timing chain 3: MEMCS* asserted to data available

Time from $t_{1SYSCLK\uparrow}$ to $t_{2SYSCLK\uparrow}$ = 100 ns

Margin 7 (M 65656) (MEC) =
= ($t_{2SYSCLK\uparrow}$ - $t_{1SYSCLK\uparrow}$) - mec:t10 - ram:trELQV - ac245:tp - mec:t71
= 100 - 15 - 45 - 10.5 - 3 = 26.5 ns

Margin 8 (M 65608) (MEC) =
= ($t_{2SYSCLK\uparrow}$ - $t_{1SYSCLK\uparrow}$) - mec:t10 - ram:trELQV - ac245:tp - mec:t71
= 100 - 15 - 25 - 10.5 - 3 = 46.5 ns

Margin 9 (M 65656) (IU) =
= ($t_{2SYSCLK\uparrow}$ - $t_{1SYSCLK\uparrow}$) - mec:t10 - ram:trELQV - ac245:tp - iu:tDIS
= 100 - 15 - 45 - 10.5 - 7 = 22.5 ns

Margin 10 (M 65608) (IU) =
= ($t_{2SYSCLK\uparrow}$ - $t_{1SYSCLK\uparrow}$) - mec:t10 - ram:trELQV - ac245:tp - iu:tDIS
= 100 - 15 - 25 - 10.5 - 7 = 42.5 ns

Timing chain 4: Address stable to data available

Time from $t_{1SYSCLK\uparrow}$ to $t_{2SYSCLK\uparrow}$ = 100 ns

Margin 11 (M 65656) (MEC) =
= ($t_{2SYSCLK\uparrow}$ - $t_{1SYSCLK\uparrow}$) - ac377:tpCkQ - ram:trAVQV - ac245:tp - mec:t71
= 100 - 11 - 45 - 10.5 - 3 = 30,5 ns

Margin 12 (M 65608) (MEC) =
= ($t_{2SYSCLK\uparrow}$ - $t_{1SYSCLK\uparrow}$) - ac377:tpCkQ - ram:trAVQV - ac245:tp - mec:t71
= 100 - 11 - 25 - 10.5 - 3 = 50,5 ns

Margin 13 (M 65656) (IU) =
= ($t_{2SYSCLK\uparrow}$ - $t_{1SYSCLK\uparrow}$) - ac377:tpCkQ - ram:trAVQV - ac245:tp - iu:tDIS
= 100 - 11 - 45 - 10.5 - 7 = 26,5 ns

Margin 14 (M 65608) (IU) =
= ($t_{2SYSCLK\uparrow}$ - $t_{1SYSCLK\uparrow}$) - ac377:tpCkQ - ram:trAVQV - ac245:tp - iu:tDIS
= 100 - 11 - 25 - 10.5 - 7 = 46,5 ns

Lowest margin is 2,5 ns (Margin 5).However, this is the worst case figure calculated for military temperature range. In laboratory environment a typical value for ac245:tp is 6 ns instead of 10,5 ns which is the worst case value, giving a margin of 7 ns.

4.3.1.2. Data Hold Time to IU and MEC During Load

Data shall be available at $t_{2SYSCLK\uparrow}$ with a hold time of mec:t78 = 1 ns.
Data shall be available at $t_{2SYSCLK\uparrow}$ with a hold time of iu:tDIH =9 ns.

A hold time on the RAM data bus of t_{cap} 5 ns resulting from capacitive loads is assumed.

Timing chain 1: MEMBEN* deasserted to data invalid

Worst case when SYSCLK output delay mec:t69 is maximum (20 ns), ac245:tp is minimum (1 ns) and MEMBEN* output delay is minimum.

Margin 1 (MEC) =
= mec:t13 + ac245:tp - mec:t69 - mec:t78 =
= mec:t13 + 1 - 20 - 1
To keep margin 1 > 0, MEMBEN* output delay (mec:t13) must be > 20 ns . A more realistic value for ac245:tp is 6 ns which change the condition on mec:t13 to be > 14 ns.
The maximum value for mec:t13 is 35 ns.

Margin 2 (IU) =
= mec:t13 + ac245:tp - mec:t69 - iu:tDIH =
= mec:t13 + 1 - 20 - 9
To keep margin 2 > 0, MEMBEN* output delay (mec:t13) must be > 28 ns .
The maximum value for mec:t13 is 35 ns.

Timing chain 2: OE* deasserted to data invalid

Worst case when SYSCLK output delay mec:t69 is maximum (20 ns), ac245:tp is minimum (1 ns) and OE* output delay is minimum.

Margin 1 (M 65656) (MEC) =
= mec:t16 + ram:trGHQZ + t_{cap} + ac245:tp - mec:t69 - mec:t78
= mec:t16 + 15 + 5 + 6 - 20 - 1
To keep margin 1 > 0, OE* output delay (mec:t16) must be > -5 ns .
The maximum value for mec:t16 is 35 ns.

Margin 2 (M 65608) (MEC) =
= mec:t16 + ram:trGHQZ + t_{cap} + ac245:tp - mec:t69 - mec:t78
= mec:t16 + 8 + 5 + 6 - 20 - 1
To keep margin 2 > 0, OE* output delay (mec:t16) must be > 2 ns .
The maximum value for mec:t16 is 35 ns.

Margin 3 (M 65656) (IU) =
= mec:t16 + ram:trGHQZ + t_{cap} + ac245:tp - mec:t69 - iu:tDIH
= mec:t16 + 15 + 5 + 6 - 20 - 9
To keep margin 3 > 0, OE* output delay (mec:t16) must be > 3 ns .
The maximum value for mec:t16 is 35 ns.

Margin 4 (M 65608) (IU) =
= mec:t16 + ram:trGHQZ + t_{cap} + ac245:tp - mec:t69 - iu:tDIH
= mec:t16 + 8 + 5 + 6 - 20 - 9
To keep margin 4 > 0, OE* output delay (mec:t16) must be > 10 ns .
The maximum value for mec:t16 is 35 ns.

This analys shows that, in some cases, the data hold time can not be guaranteed for the worst case when the SYSCLK output delay is maximum (t69 = 20 ns).
Lowest margin is when MEMBEN* output delay > 28 ns

However, a realistic assumption is that the timing of mec:t69 and mec:t16 is correlated so that if t69 is maximum then t16 is near maximum, giving more hold time than worst-worst case.

4.3.1.3. Data Setup Time to MEC During Store

Data shall be available at $t_{4SYSCLK\downarrow}$ with a setup time of mec:t19 =20 ns.
iu:tpDClkB = iu:t_{DOD} = 35 ns (D<31:0> output delay)

Timing chain 1: IU store data available
Time from $t_{2SYSCLK\downarrow}$ to $t_{4SYSCLK\downarrow}$ = 200 ns
Margin 1 = $(t_{3SYSCLK\downarrow} - t_{2SYSCLK\downarrow})$ - iu:tpDClkB - mec:t19
= 200 - 35 - 20 = 145 ns

The margin is +145 ns.

4.3.1.4. Data Setup Time to RAM During Store

Data shall be available at MEMWR1* deasserted with a data setup time of ram:twDVWH

Timing chain 1: IU store data available
Time from $t_{2SYSCLK\downarrow}$ to $t_{4CLK2\downarrow 1}$ = 175 ns - mec:t69 = 175 - 20 = 155 ns (Worst case when SYSCLK output delay is maximum).

Margin 1 (M 65656) =
= $(t_{4CLK2\downarrow 1} - t_{2SYSCLK\downarrow})$ + mec:t15 - iu:tpDClkB - ram:twDVWH =
= 155 + mec:t15 - 35 - 25 = (worst case when mec:t15 is 0) =95 ns
Note that MEMWR1* output delay (mec:t15) has the maximun value 35 ns.
Margin 1 = mec:t15+ 95 ns.

Margin 2 (M 65608) =
= $(t_{4CLK2\downarrow 1} - t_{2SYSCLK\downarrow})$ + mec:t15 - iu:tpDClkB - ram:twDVWH =
= 155 + mec:t15 - 35 - 15 = (worst case when mec:t15 is 0) =105 ns
Note that MEMWR1* output delay (mec:t15) has the maximun value 35 ns.
Margin 2 = mec:t15 +105 ns.

4.3.1.5. Data Hold Time to MEC During Store

Data shall be available at $t_{4SYSCLK\downarrow}$ with a hold time of mec:t20 = 1 ns

Timing chain 1: IU store data invalid
IU data output is valid from$t_{4SYSCLK\downarrow}$ for a minimum of 4 ns which is more than 1 ns.
iu:tDOH = iu:tpDCkBI > mec:t20 =>

The margin is > 3 ns.

4.3.1.6. Data Hold Time to RAM During Store

Data shall be available at MEMWR1* (see paragraph 3.3.1.1.4.~~3.3.1.1.4.~~) deasserted with a hold time of ram:twWHDX

Timing chain 1: IU store data invalid
Time from $t_{4CLK2\downarrow 1}$ to $t_{4SYSCLK\downarrow}$ = 25 ns + mec:t69 = 25 ns (worst case)

Margin 1 (M 65656 and M 65608) =
= ($t_{4SYSCLK\downarrow}$ - $t_{4CLK2\downarrow 1}$) + iu:tpDCkBI - mec:t15 - ram:twWHDX =
= 25 + 4 - 35 - 0 = -6 ns

This is worst case when:

1. IU data output valid has the minimum value 4 ns
2. SYSCLK output delay is 0 ns
3. MEMWR1* output delay has the maximum value 35 ns

However, a realistic assumption is that the timing of mec:t69 and mec:t15 is correlated so that if t15 is maximum then t69 is near maximum, giving more hold time than worst-worst case.

4.3.1.7. Check Bits Setup Time to RAM During Store

Check bit data shall be available at MEMWR1* (see paragraph 3.3.1.1.4.~~3.3.1.1.4.~~) deasserted with a setup time of ram:twDVWH (25 ns for M65656 and 15 ns for M65608)

Timing chain 1: Check bit data available
Time from $t_{3SYSCLK\uparrow}$ to $t_{4CLK2\downarrow 1}$ = 125 ns - mec:t69 = 105 ns (worst case)
For the worst case:
1. The SYSCLK output delay is maximum (mec:t69 = 20)
2. MEMWR1* is deasserted without any delay (mec:t15 = 0)
3. CB output delay has the maximum value of 35 ns (mec:t23 = 35)

Margin 1 (M 65656) =
= ($t_{4CLK2\downarrow 1}$ - $t_{3SYSCLK\uparrow}$) + mec:t15 - mec:t23 - ram:twDVWH =
= 105 + 0 - 35 - 25 = 45 ns

Margin 2 (M 65608) =
= ($t_{4CLK2\downarrow 1}$ - $t_{3SYSCLK\uparrow}$) + mec:t15 - mec:t23 - ram:twDVWH =
= 105 + 0 - 35 - 15 = 55 ns

The CB setup time to RAM is guaranteed for both devices.

4.3.1.8. Check Bits Hold Time to RAM During Store

Check bit data shall be available at MEMWR1* (see paragraph 3.3.1.1.4.~~3.3.1.1.4.~~) deasserted with a hold time of ram:twWHDX (0 ns).
For the worst case:
1. The SYSCLK output delay is minimum (mec:t69 = 0)

2. MEMWR1* is deasserted with a maximum delay (mec:t15 = 35)
3. CB output valid to high Z delay has the minimum value of 2 ns (mec:t24 = 2)

Timing chain 1: CB store data invalid
Time from $t_{4CLK2\downarrow 1}$ to $t_{5SYSCLK\uparrow}$ = 75 ns + mec:t69 = 75 ns (worst case)

Margin 1 (M 65656 and M65608)) =
= ($t_{5SYSCLK\uparrow}$ - $t_{4CLK2\downarrow 1}$) + mec:t24 - mec:t15 - ram:twWHDX =
= 75 + 2 - 35 - 0 = 42 ns

According to this analys, the CB hold time to RAM is guaranteed foboth memory types.

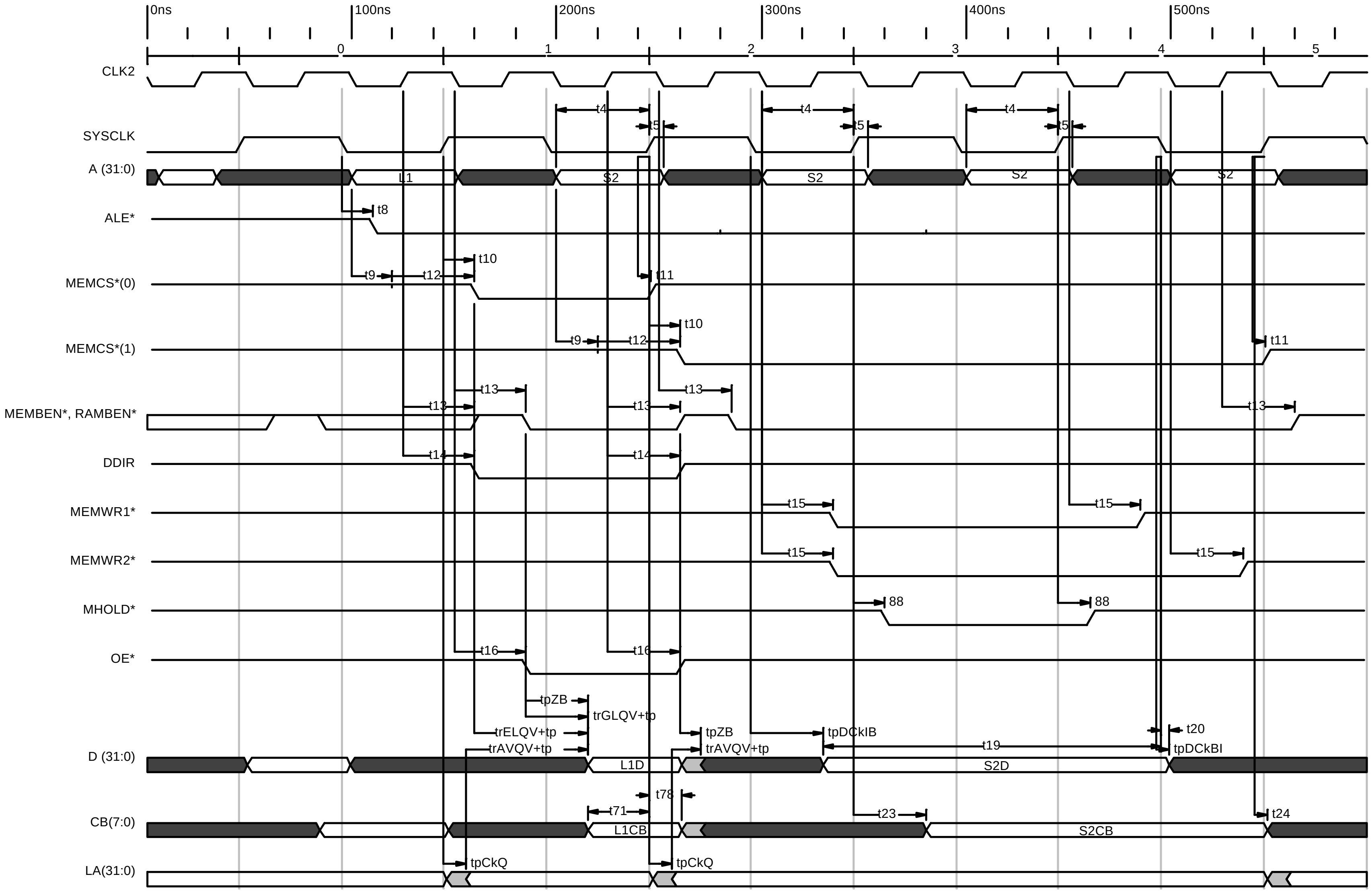


Figure 14~~14~~ - DEM32 RAM Load at 0Ws and Store at 1 Ws Sequence.

4.3.2. EEPROM Access

The EEPROM access timing is determined by the IU, MEC, buffer and memory timing. In Figure 15 the timing is shown for a load operation at 1 waitstate, 10 MHz system clock. As byte mode access is used each word fetch takes a total of 4*2+2=10 cycles.

4.3.2.1. Data Setup Time to MEC During Load

Data shall be available at $t_{3SYSCLK\uparrow}$ with a setup time of mec:t71 = 3 ns.

Timing chain 1: MEMBEN* asserted to data available
Time from $t_{1CLK2\downarrow 1}$ to $t_{3SYSCLK\uparrow}$ = 175 ns + mec:t69 = 175 ns (worst case)
Margin 1 = $(t_{3SYSCLK\uparrow} - t_{1CLK2\downarrow 1})$ - mec:t13 - ac245:tpZB - mec:t71
= 175 - 35 - 10.5 - 3 = 126.5 ns

Timing chain 2: OE* asserted to data available
Time from $t_{1CLK2\downarrow 1}$ to $t_{3SYSCLK\uparrow}$ = 175 ns + mec:t69 = 175 ns (worst case)
Margin 2 (M 65656) =
= $(t_{3SYSCLK\uparrow} - t_{1CLK2\downarrow 1})$ - mec:t16 - ram:trGLQV - ac245:tp - mec:t71 =
= 175 - 35 - 20 - 10.5 - 3 = 106.5 ns

Margin 3 (M 65608) =
= $(t_{3SYSCLK\uparrow} - t_{1CLK2\downarrow 1})$ - mec:t16 - ram:trGLQV - ac245:tp - mec:t71 =
= 175 - 35 - 10 - 10.5 - 3 = 116.5 ns

Timing chain 3: ROMCS* asserted to data available
Time from $t_{2SYSCLK\uparrow}$ to $t_{3SYSCLK\uparrow}$ = 100 ns
Margin 4 (M 65656) =
= $(t_{3SYSCLK\uparrow} - t_{2SYSCLK\uparrow})$ - mec:t10 - ram:trELQV - ac245:tp - mec:t71 =
= 100 - 15 - 45 - 10.5 - 3 = 26.5 ns

Margin 5 (M 65608) =
= $(t_{3SYSCLK\uparrow} - t_{2SYSCLK\uparrow})$ - mec:t10 - ram:trELQV - ac245:tp - mec:t71 =
= 100 - 15 - 25 - 10.5 - 3 = 46.5 ns

Timing chain 4: Address stable to data available
Time from $t_{1SYSCLK\uparrow}$ to $t_{3SYSCLK\uparrow}$ = 200 ns
Margin 6 (M 65656) =
= $(t_{3SYSCLK\uparrow} - t_{1SYSCLK\uparrow})$ - ac377:tpCkQ - ram:trAVQV - ac245:tp - mec:t71 =
= 200 - 11 - 45 - 10.5 - 3 = 130.5 ns

Margin 7 (M 65608) =
= $(t_{3SYSCLK\uparrow} - t_{1SYSCLK\uparrow})$ - ac377:tpCkQ - ram:trAVQV - ac245:tp - mec:t71 =
= 200 - 11 - 25 - 10.5 - 3 = 150.5 ns

Lowest margin is 26,5 ns

4.3.2.2. Data Setup Time to IU During Load

Data shall be available at $t_{6SYSCLK\uparrow}$ with a setup time of iu:tDIS =7 ns.

Timing chain 1: Data available from MEC to IU
Margin 1 = $(t_{6SYSCLK\uparrow} - t_{5SYSCLK\downarrow})$ - mec:t21 - iu:tDIS =
= 50 - 20 - 6 = 24 ns

The margin is 24 ns

4.3.2.3. Data Hold Time to MEC During Load

Data shall be available at $t_{5SYSCLK\uparrow}$ with a hold time of mec:t22 = 2 ns.

Timing chain 1: MEMBEN* deasserted to data invalid
Margin 1 = ac245:tpBZ - mec:t22

The margin is 0 ns

4.3.2.4. Data Hold Time to IU During Load

Data shall be available at $t_{6SYSCLK\uparrow}$ with a hold time of iu:tDIH =9 ns.

Timing chain 1: $t_{6SYSCLK\uparrow}$ to MEC data invalid
Margin 1 = mec:t22 - iu:tDIH

The margin is -7 ns!However, at room temperature mec:t22 is considerably larger than 2 ns (worst case value) giving a sufficient margin even if considering worst case hold time on the IU.

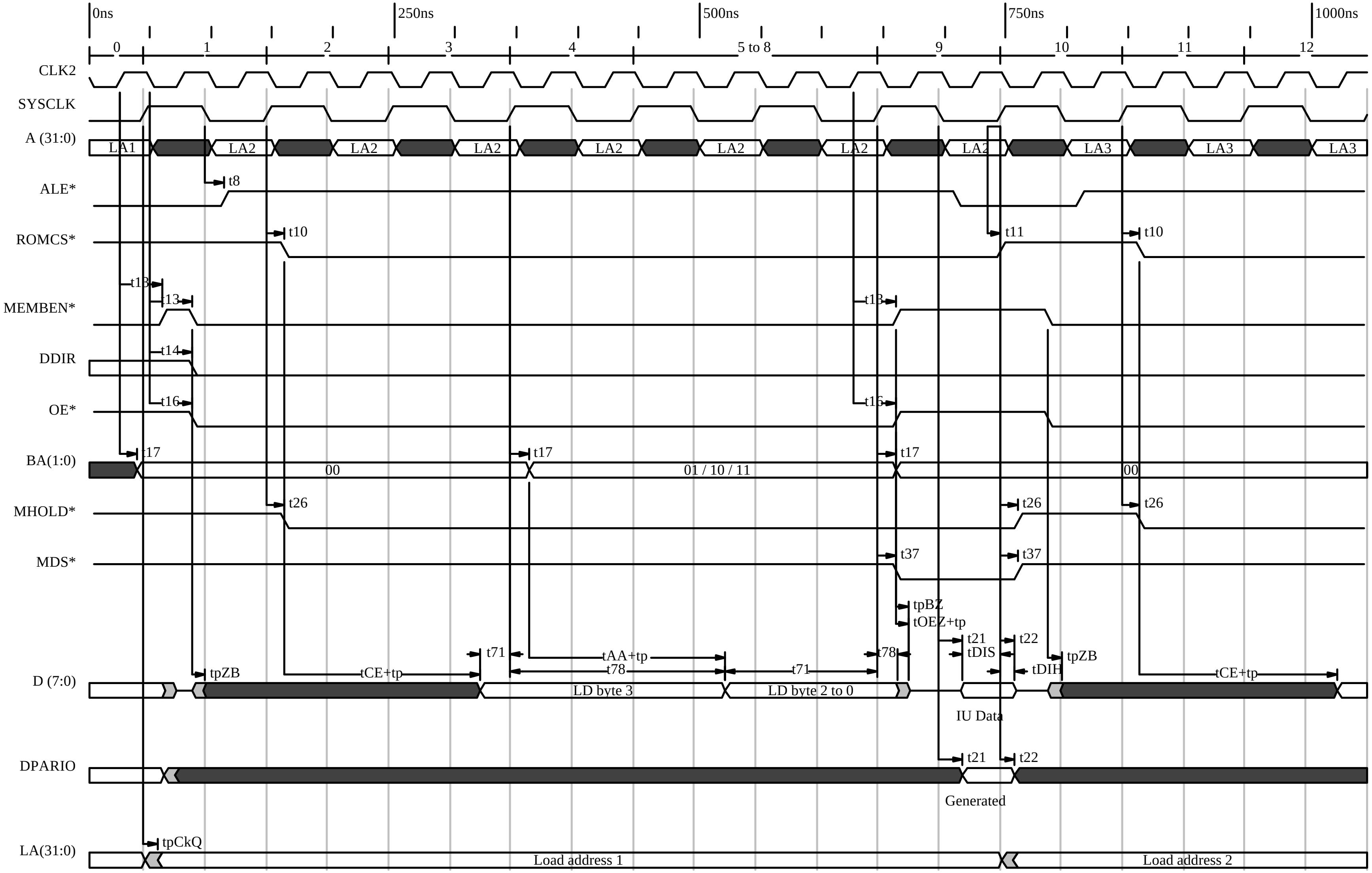


Figure 15~~15~~ - DEM32 PROM Load at 1 WS.

4.3.3. I/O (ATAC) Access

The ATAC is addressed as I/O unit 0 in the DEM32. As the ATAC cannot be connected to the ERC32 system bus directly, handshaking logic must be implemented. In the DEM32 the handshaking is implemented in a PLD, 22V10 (see Appendix 1). The timing is thus largely controlled by the PLD.

4.3.3.1. Data Setup Time to IU and MEC During Load

Data shall be available at $t_{7SYSCLK\uparrow}$ with a setup time of mec:t71 = 3 ns.
Data shall be available at $t_{7SYSCLK\uparrow}$ with a setup time of iu:tDIS = 6 ns.

The analysis is trivial since the PLD handshake logic ensures a setup time of more than four SYSCLK cycles.

4.3.3.2. Data Hold Time to IU and MEC During Load

Data shall be available at $t_{7SYSCLK\uparrow}$ with a hold time of mec:t78 = 1 ns.
Data shall be available at $t_{7SYSCLK\uparrow}$ with a hold time of iu:tDIH =9 ns.

A hold time on the I/O data bus of t_{cap} 5 ns resulting from capacitive loads is assumed.

Timing chain 1: ATAC_CE* deasserted to data invalid
Margin 1 = pld22v10:tp + t_{cap} - mec:t78 = 4 + 5 - 1 = 8 ns
Margin 2 = pld22v10:tp + t_{cap} - iu:tDIH = 4 + 5 - 9 = 0 ns

The margin is 0 ns.

4.3.3.3. Data Setup Time to ATAC During Store

Data shall be available at $t_{4SYSCLK\uparrow}$ with a setup time of ATAC:tDIS = 0 ns

Timing chain 1: IU store data available
Time from $t_{2SYSCLK\downarrow}$ to $t_{4SYSCLK\uparrow}$ = 150 ns
Margin 1 = ($t_{4SYSCLK\uparrow}$ - $t_{2SYSCLK\downarrow}$)- iu:tpDClkB - ac245:tp =
= 150 - 35 - 10,5 = 104,5

The margin is 104,5 ns.

4.3.3.4. Data Hold Time to ATAC During Store

Data shall be available at $t_{4SYSCLK\uparrow}$ with a hold time of atac:tDIH

The analysis is trivial since the PLD handshake logic ensures a hold time of more than three SYSCLK cycles.

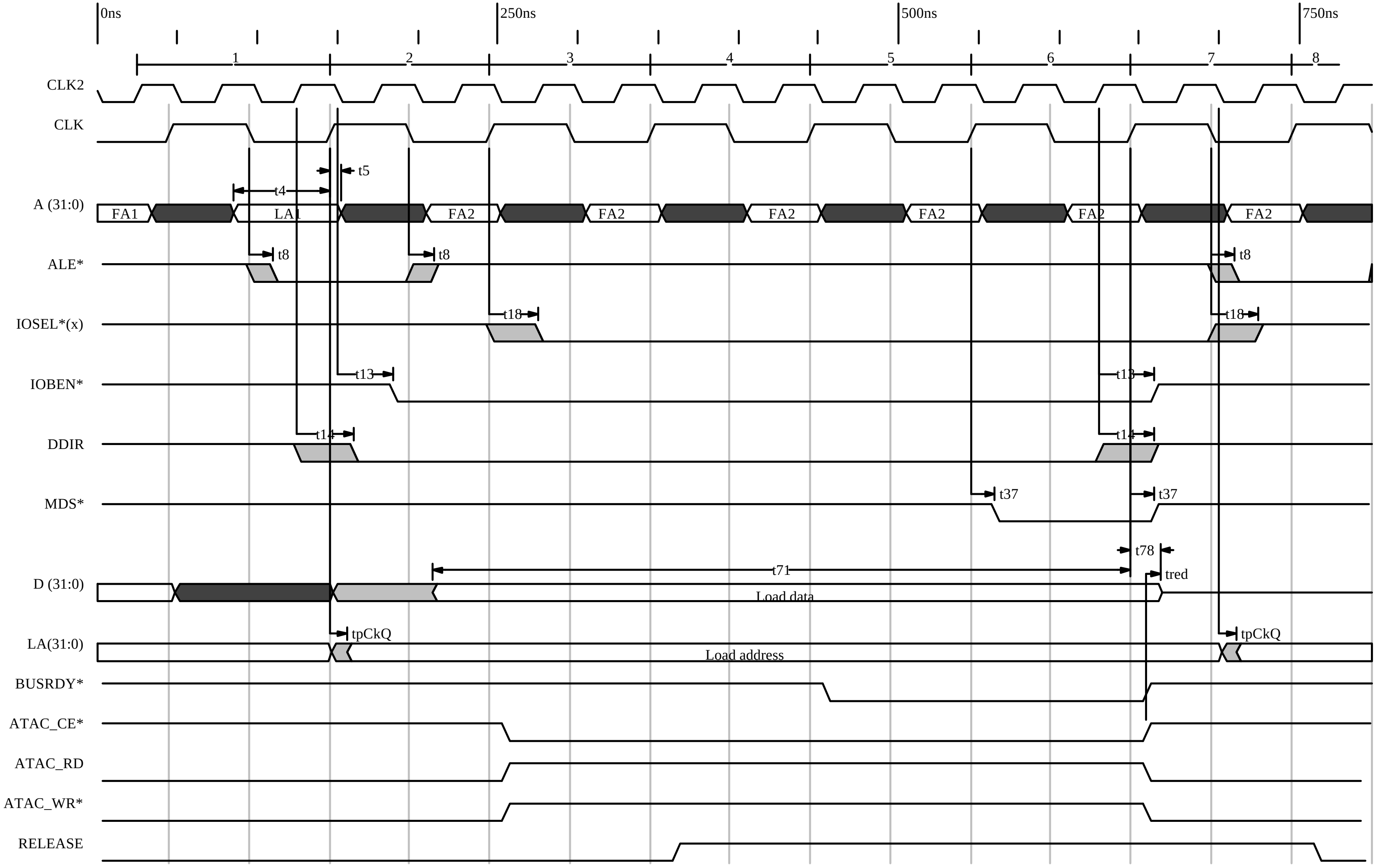


Figure 16~~16~~ - DEM32 IO Load ATAC.

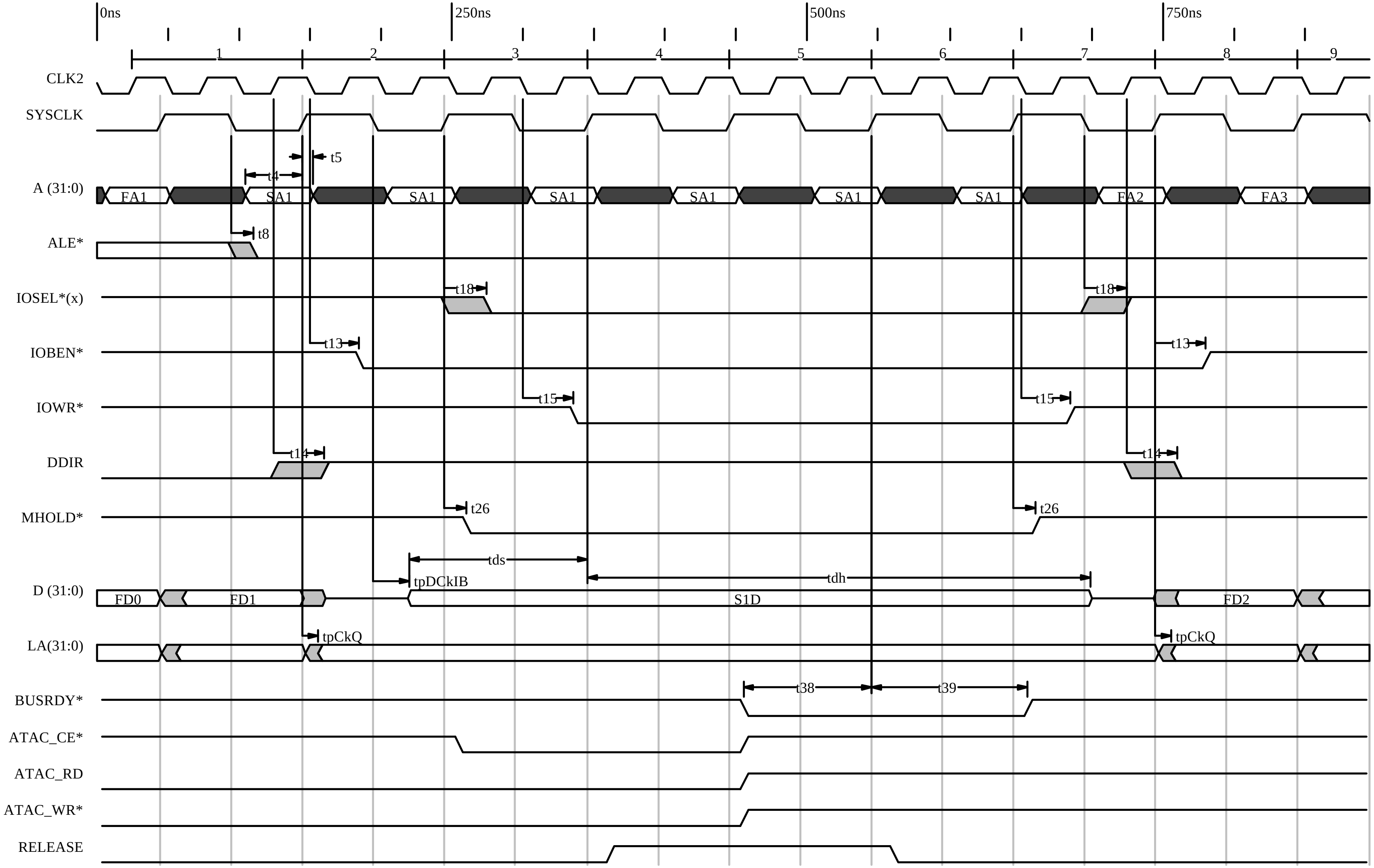


Figure 17~~17~~ - DEM32 IO Store ATAC.

APPENDIX 1 - PLD EQUATIONS

ATACIF - IC 43

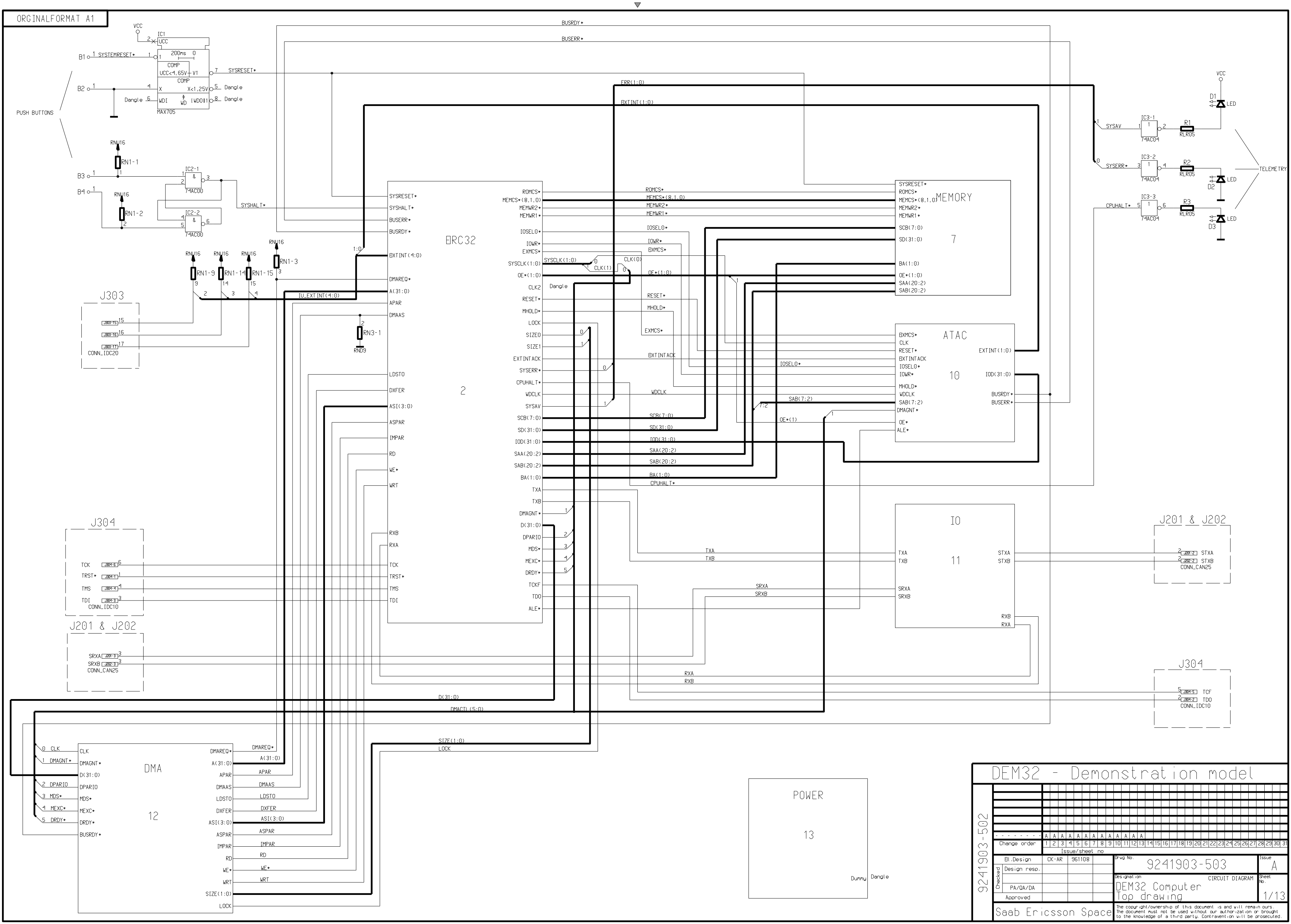
```
MODULE ic43
"
" Copyright SAAB ERICSSON SPACE AB
"
"---------------------------------------------------------------------------------------------
"
" The ownership and copyright of this document belong to
" Saab Ericsson Space AB and it must not be disclosed, copied, altered
" or used without the written permission of Saab Ericsson Space AB.
"
"---------------------------------------------------------------------------------------------
"
  TITLE                         'Ada Tasking Accelerator Coprocessor Interface'
" File name:                    ATACIF.ABL
" ABEL unit:                    (Type)
" Purpose and functionality:    (Text)
" Reference:                    (RDx)
" Analysis Dependencies:        (N/A)
" Limitations:                  (N/A)
" Fidelity:                     (N/A)
" Discrepancies:                (N/A)
" Usage:                        (N/A)
" I/O:                          (N/A)
" Operations:                   (N/A)
" Assertions:                   (N/A)
" Development Platform:         (N/A)
" Analyzer:                     (No Dependencies)
" Synthesis:                    (No Dependencies)
"
"---------------------------------------------------------------------------------------------
"
" Revision history:             (all revisions included)
"
"---------------------------------------------------------------------------------------------
" Version No:    Author:                                Modification Date:  Changes made:
"---------------------------------------------------------------------------------------------
"       v1.0     Jörgen Jost auf der Stroth             940428              New issue
"       v1.1     Jörgen Jost auf der Stroth   940915                 Added signal BUSERR_N
"       v1.1     Jörgen Jost auf der Stroth   940920                 Removed signal ALE_N
"                                                                              Logical behavior and START
"                                                                              replaced by IOBEN_N
"                                                       950119                 PIN Placement
"       v2.0    Ola Persson                     951114              Statemachine rewritten
"                                                                   for DEM32 board
"---------------------------------------------------------------------------------------------
" SAAB ERICSSON SPACE AB
" Delsjomotet                   Phone Int: +46 31 35 00 00
" S-405 15 GOTHENBURG           Fax   Int: +46 31 35 95 20
" Sweden                        Telex:     27950 saabsp s
"---------------------------------------------------------------------------------------------
"
"
" This is an module that interfaces a system with ready controlled accesses to the
" ATAC (Ada Tasking Accellerator Coprocessor).
"
" device type
ic43 device 'p22v10';

                                              " Input Signals
CLK             pin 1;
IOWR_N          pin 3;
SEL_N           pin 4;
RELEASE         pin 6;

                                              " Output Signals

CE_N            pin 23 istype 'reg,pos';
RD              pin 22 istype 'reg,pos';
WR_N            pin 21 istype 'reg,pos';

BUSRDY_N        pin 20 istype 'reg,pos';

BUSERR_N        pin 19 istype 'reg,pos';

Q1              pin 18 istype 'reg,pos';
Q2              pin 17 istype 'reg,pos';

States = [Q1,Q2,CE_N];

IDLE       = [0,0,1]; " No strobes active waiting for IOSEL_N negative
READ1      = [1,1,0]; " Waiting for RELEASE from ATAC or IOSEL_N positive
READ2      = [0,1,0]; " Goto READ2
READ3      = [1,0,0]; " Goto WAIT
WRITE1     = [0,0,0]; " Waiting for RELEASE from ATAC or IOSEL_N positive
WRITE2     = [1,0,1]; " Goto WRITE3
WRITE3     = [1,1,1]; " Goto WAIT
WAIT       = [0,1,1]; " No strobes active waiting for IOSEL_N positive

equations
```

```
States.clk      = CLK;


RD = Q1.fb # Q2.fb;
WR_N =  Q1.fb # Q2.fb;
BUSRDY_N = ((States.fb == READ1) # (States.fb == WRITE1) # (States.fb == IDLE)) ;

" BUSERR is not used
BUSERR_N = 1;

state_diagram States;
state IDLE:
   if ( !SEL_N & IOWR_N ) then " Start read access
      goto READ1;
   else if ( !SEL_N & !IOWR_N ) then " Start write access
      goto WRITE1;
   else
      goto IDLE;

state READ1:
   if RELEASE then " Waiting for RELEASE from ATAC
      goto READ2;
   else if SEL_N then  " If no cycle then IDLE
      goto IDLE;
   else
      goto READ1;  " Else wait here

state READ2:
   goto READ3;

state READ3:
   goto WAIT;

state WRITE1:
   if RELEASE then " Waiting for RELEASE from ATAC
      goto WRITE2;
   else if SEL_N then  " If no cycle then IDLE
      goto IDLE;
   else
      goto WRITE1; " Else wait here

state WRITE2:
   goto WRITE3;

state WRITE3:
   goto WAIT;

state WAIT:
   if !SEL_N then
      goto WAIT;
   else
      goto IDLE;


END ic43
```

APPENDIX 2 - DEM32 SCHEMATICS

DEM32 V.3 - 92 41 903 - 503a

Schematic page [Document page]	DESIGNATION
1 [50]	DEM32 Top drawing
2 [51]	ERC32
3 [52]	IU-RT
4 [53]	FPU-RT
5 [54]	MEC
6 [55]	Buffers
7 [56]	Memory and Flash EEPROM
8 [57]	DATA SRAM
9 [58]	CB SRAM
10 [59]	ATAC
11 [60]	IO RS232
12 [61]	DMA and Test conectors
13 [62]	Power and Decoupling



DEM32 - Demonstration model

9241903-502

Change order	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
- - - - - - - - -	A	A	A	A	A	A	A	A	A	A	A	A	A																		

Issue/sheet no

El.Design	CK-AR	961108	
Checked Design resp.			
PA/QA/DA			
Approved			

Drwg No. 9241903-503

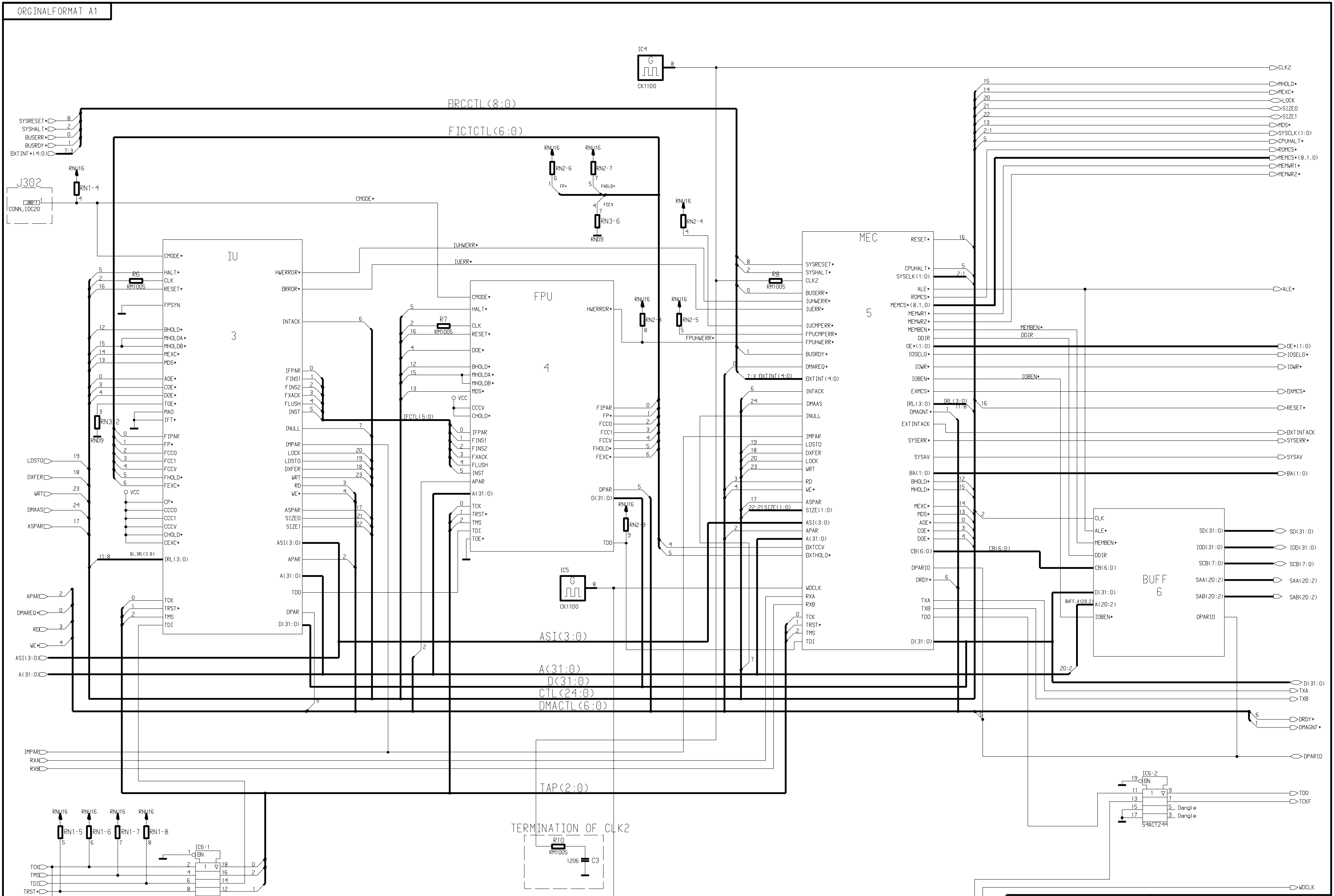
Issue A

Designation CIRCUIT DIAGRAM

DEM32 Computer
Top drawing

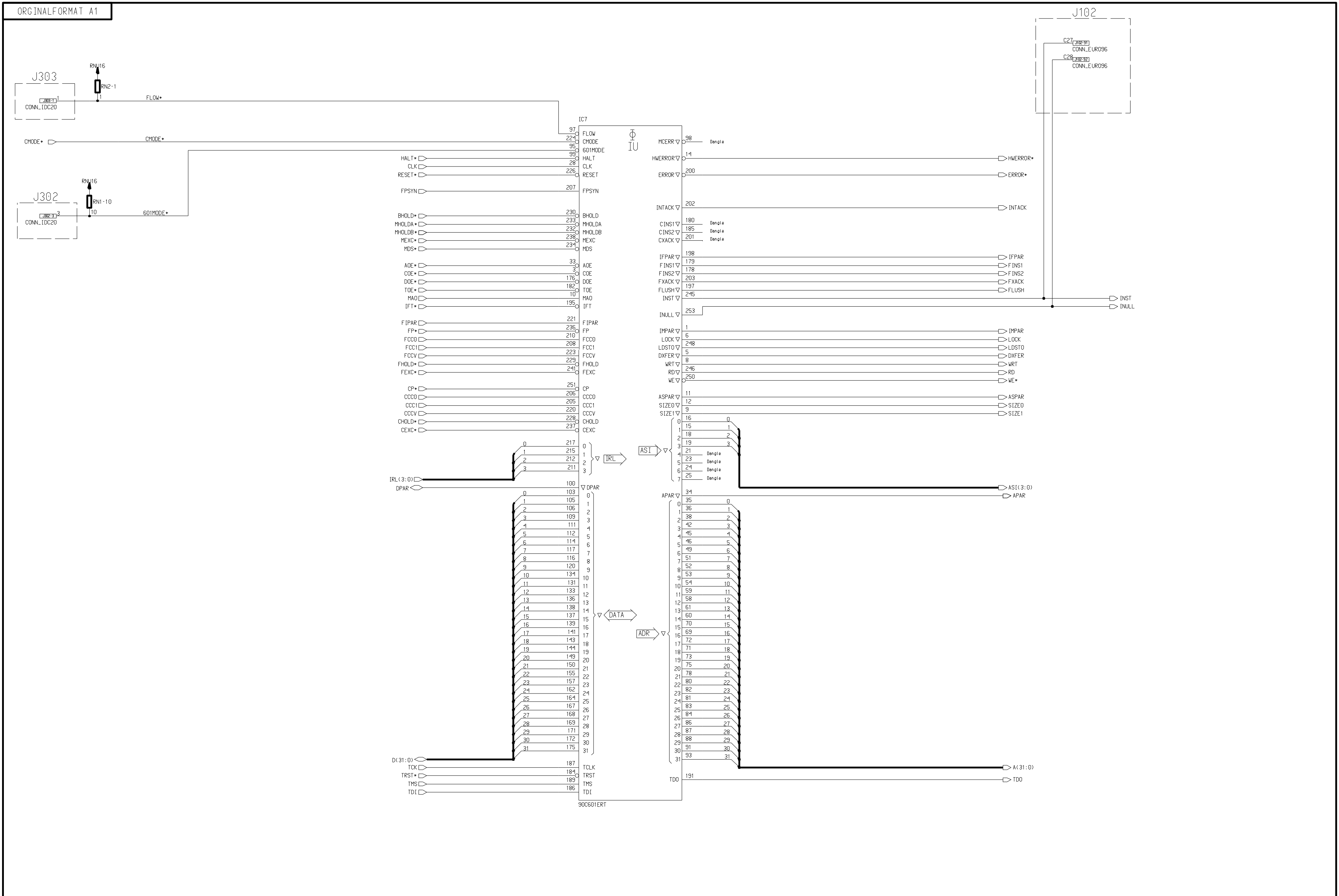
Sheet No. 1/13

Saab Ericsson Space



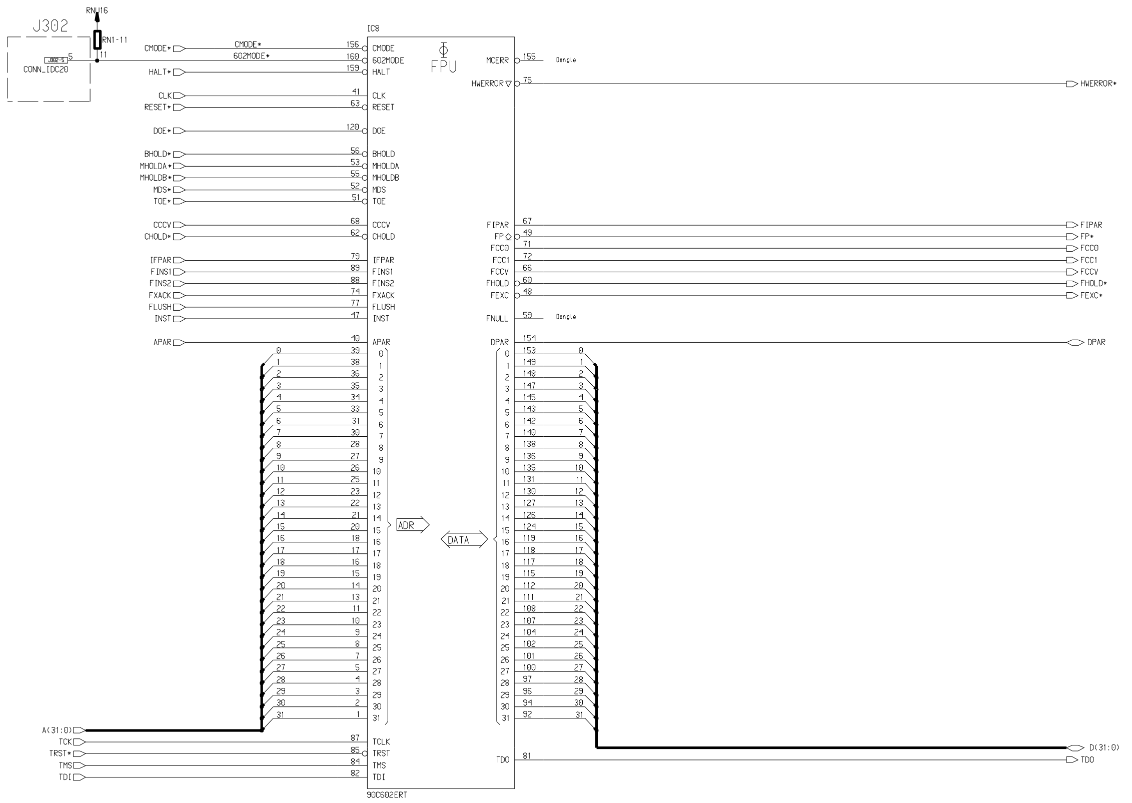
Saab Ericsson Space	Drwg No. 9241903-503	Issue A
The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.	Designation CIRCUIT DIAGRAM DEM32 ERC32	Sheet No. 2/13

ORGINALFORMAT A1



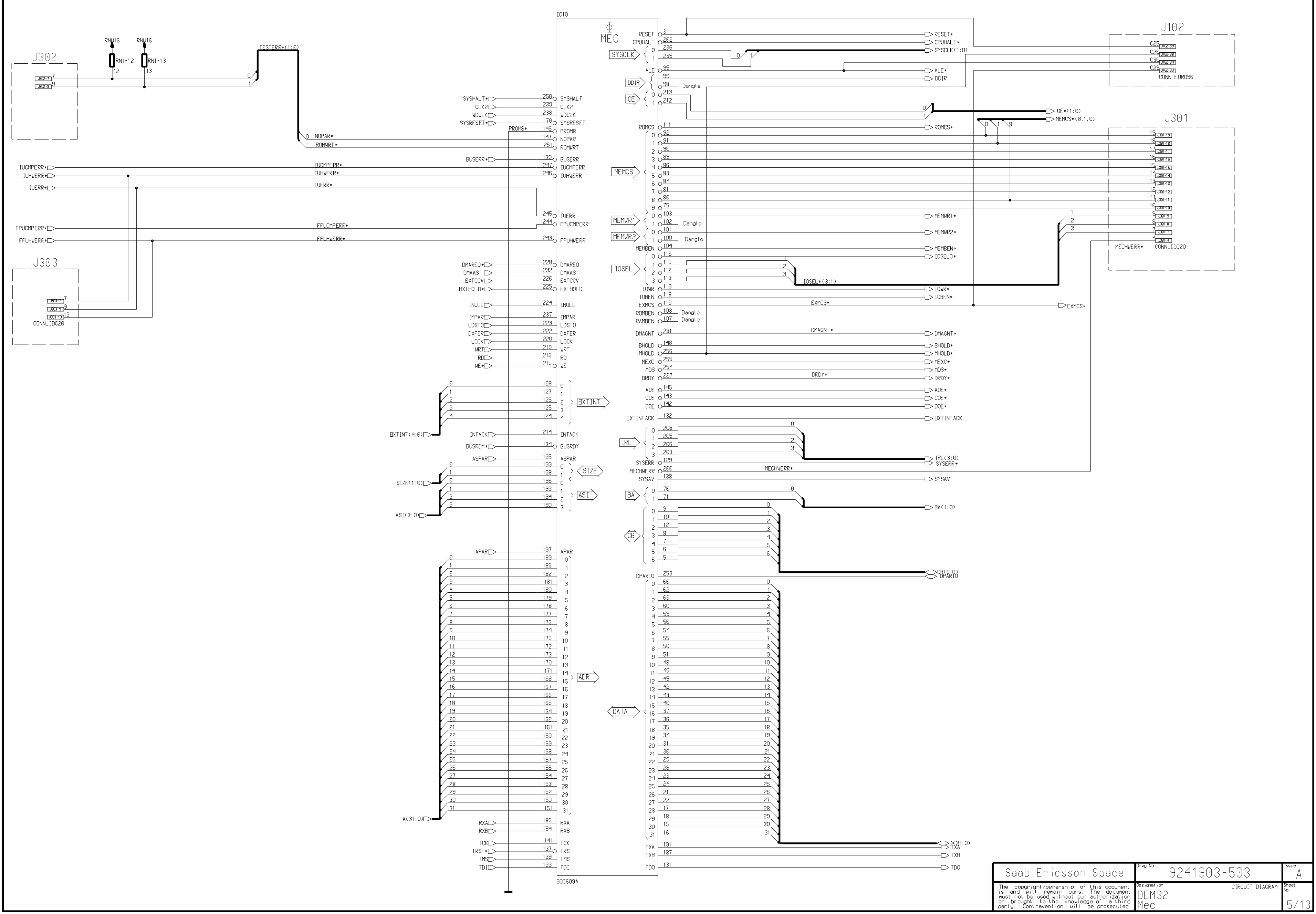
Saab Ericsson Space	Drwg No. 9241903-503	Issue A
The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.	Designation DEM32 IU-RT CIRCUIT DIAGRAM	Sheet No. 3/13

ORGINALFORMAT A1

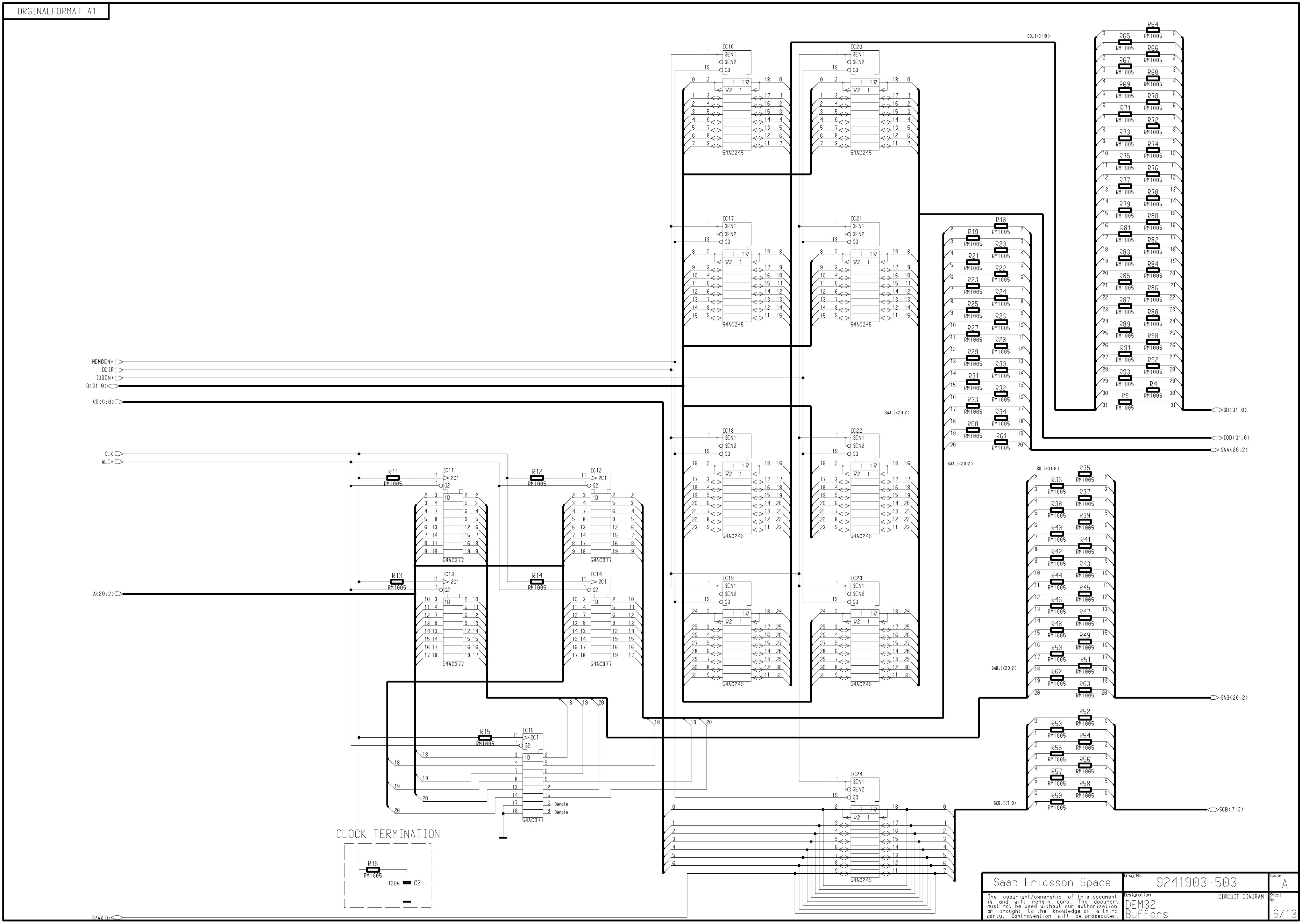


Saab Ericsson Space	Drwg No. 9241903-503	Issue A
The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.	Designation DEM32 FPU-RT CIRCUIT DIAGRAM	Sheet No. 4/13

ORGINALFORMAT A1



Saab Ericsson Space	Drwg No. 9241903-503	Issue A
The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.	Designation DEM32 Mec CIRCUIT DIAGRAM	Sheet No. 5/13



Saab Ericsson Space

The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.

Drwg No. 9241903-503

Issue A

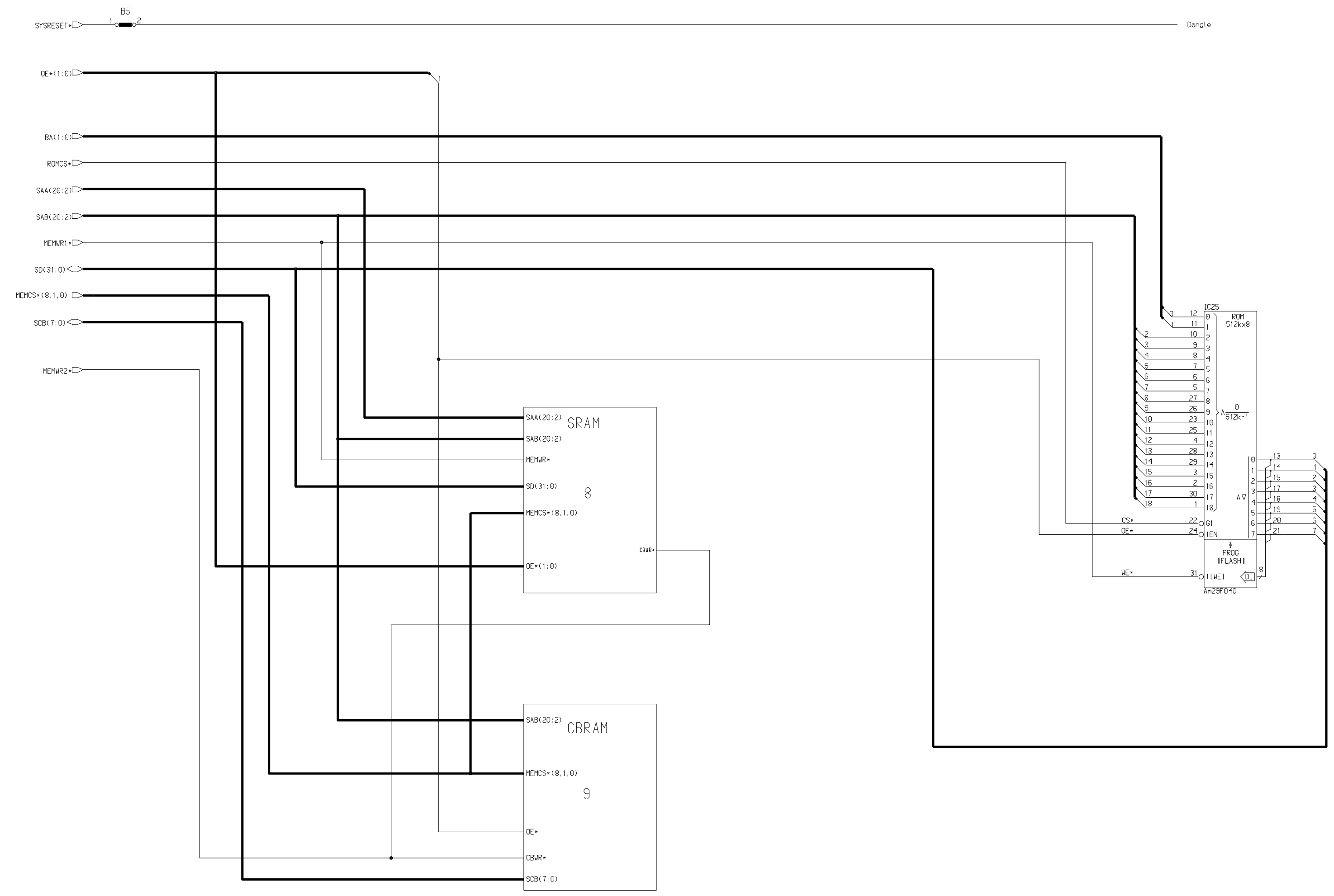
Designation CIRCUIT DIAGRAM

DEM32
Buffers

Sheet No. 6/13

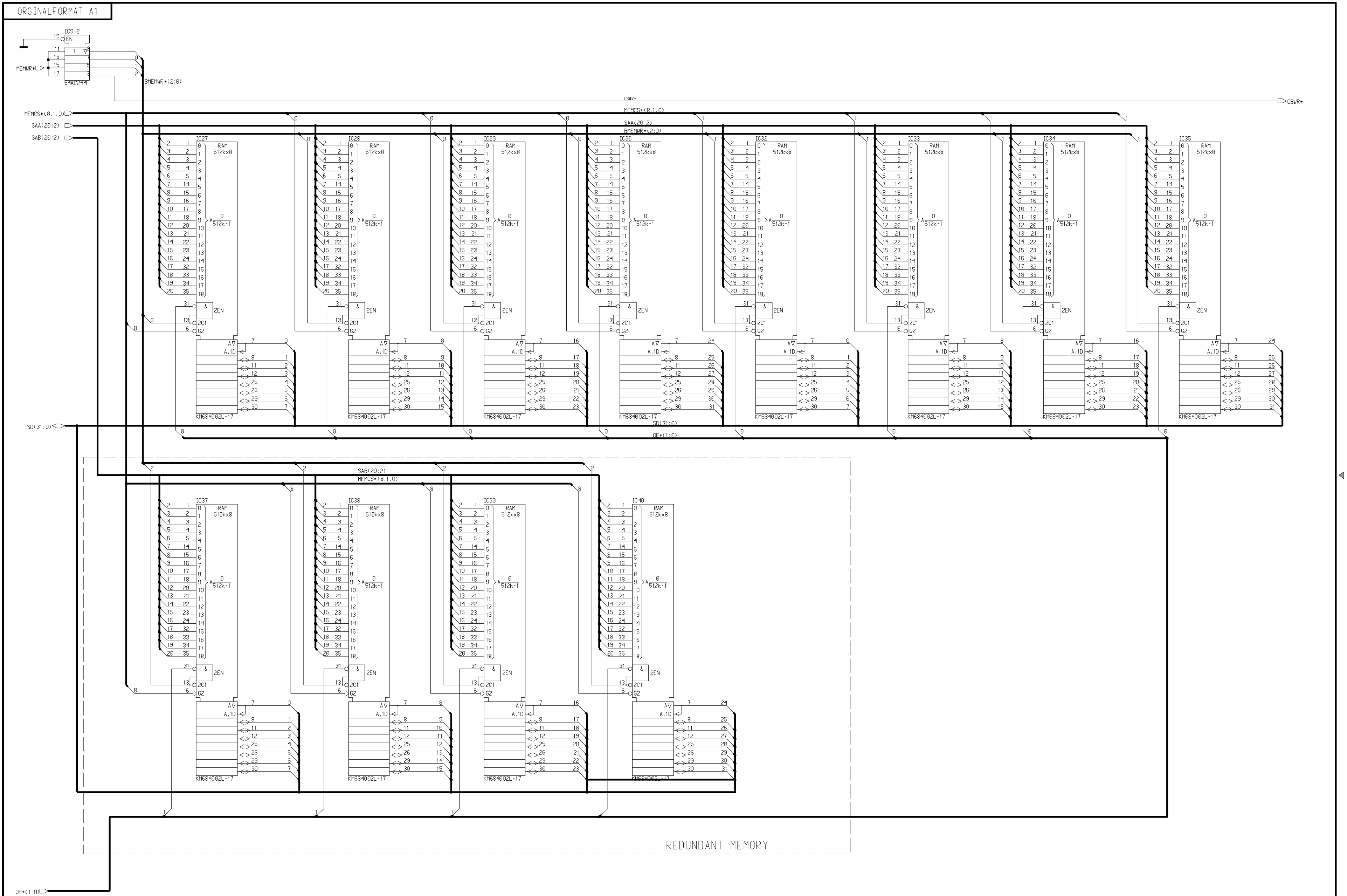
SCB_I(7:0)

ORGINALFORMAT A1



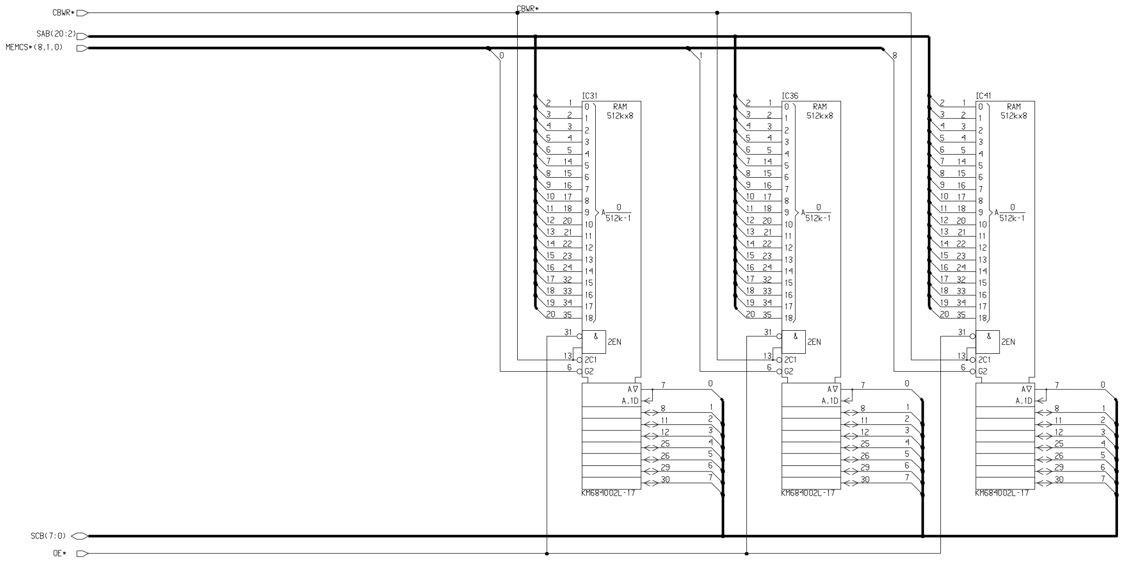
Saab Ericsson Space	Drwg No. 9241903-503	Issue A
The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.	Designation CIRCUIT DIAGRAM DEM32 Memory	Sheet No. 7/13

ORGINALFORMAT A1



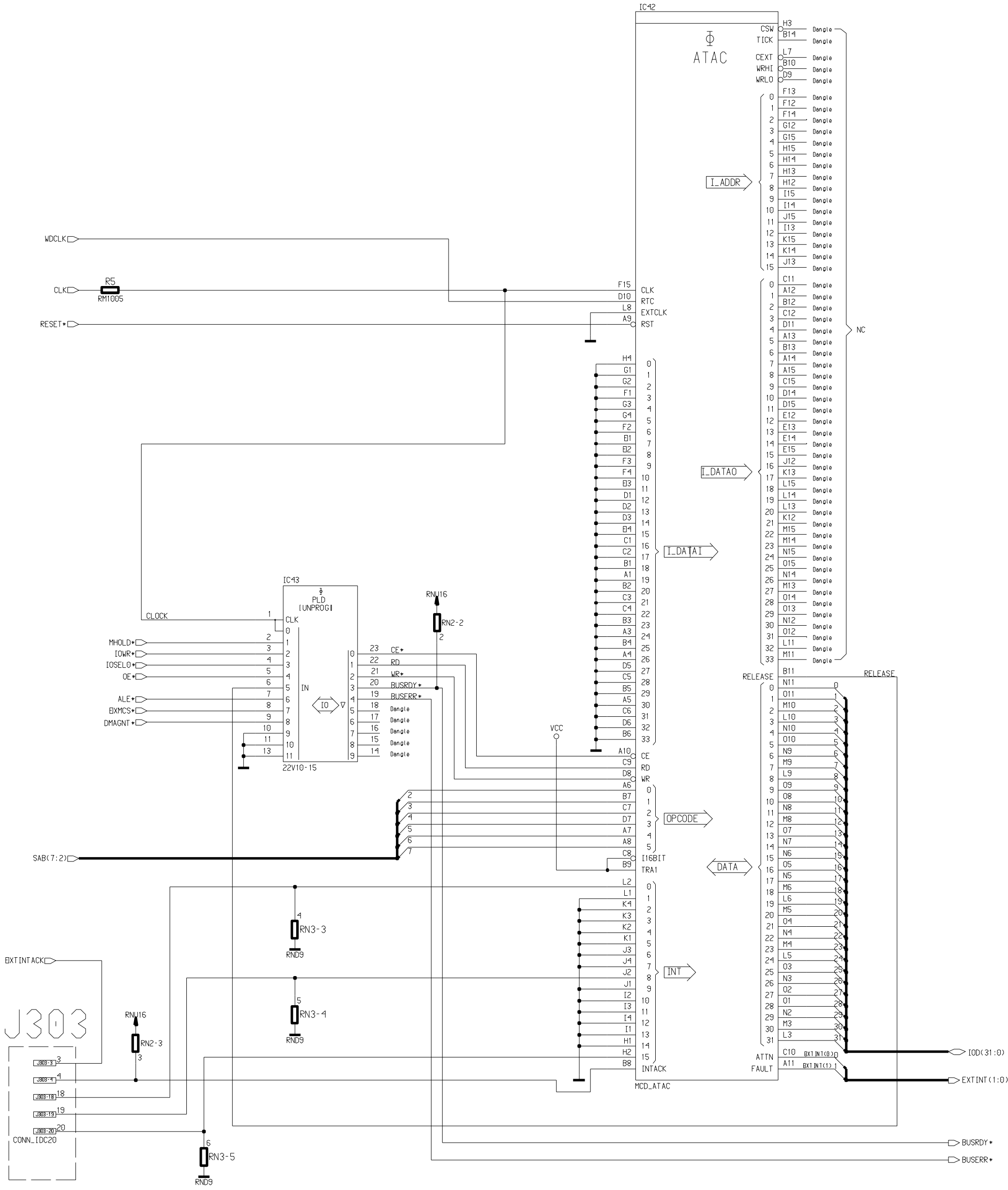
Saab Ericsson Space	Drwg No. 9241903-503	Issue A
The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.	Designation DEM32 Sram CIRCUIT DIAGRAM	Sheet No. 8/13

ORGINALFORMAT A1



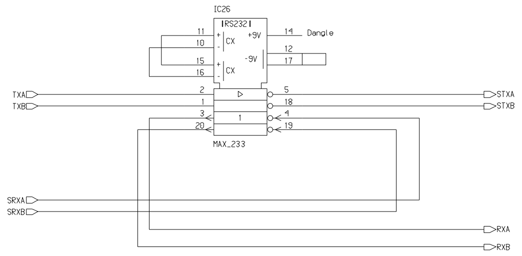
Saab Ericsson Space	Drwg No. 9241903-503	Issue A
The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.	Designation CIRCUIT DIAGRAM DEM32 CB Ram	Sheet No. 9/13

ORGINALFORMAT A1



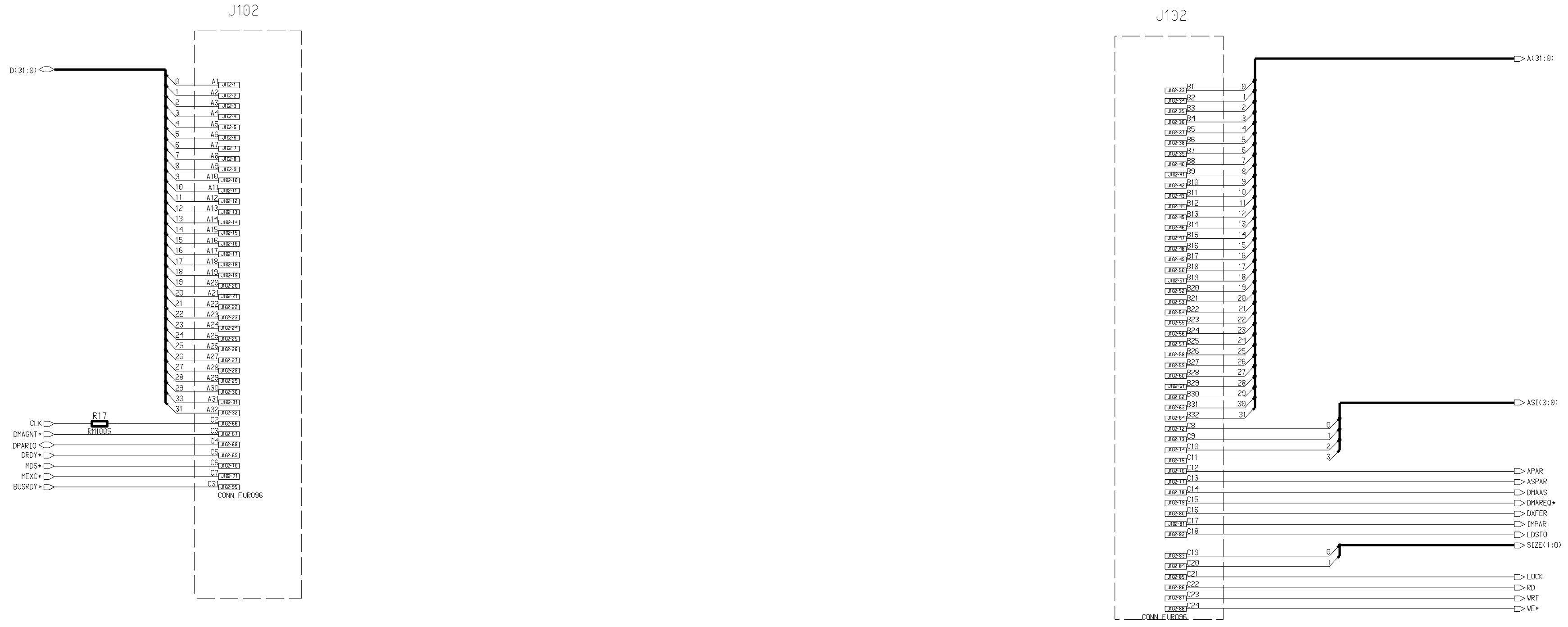
Saab Ericsson Space	Drwg No. 9241903-503	Issue A
The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.	Designation CIRCUIT DIAGRAM DEM32 Atac	Sheet No. 10/13

ORGINALFORMAT A1



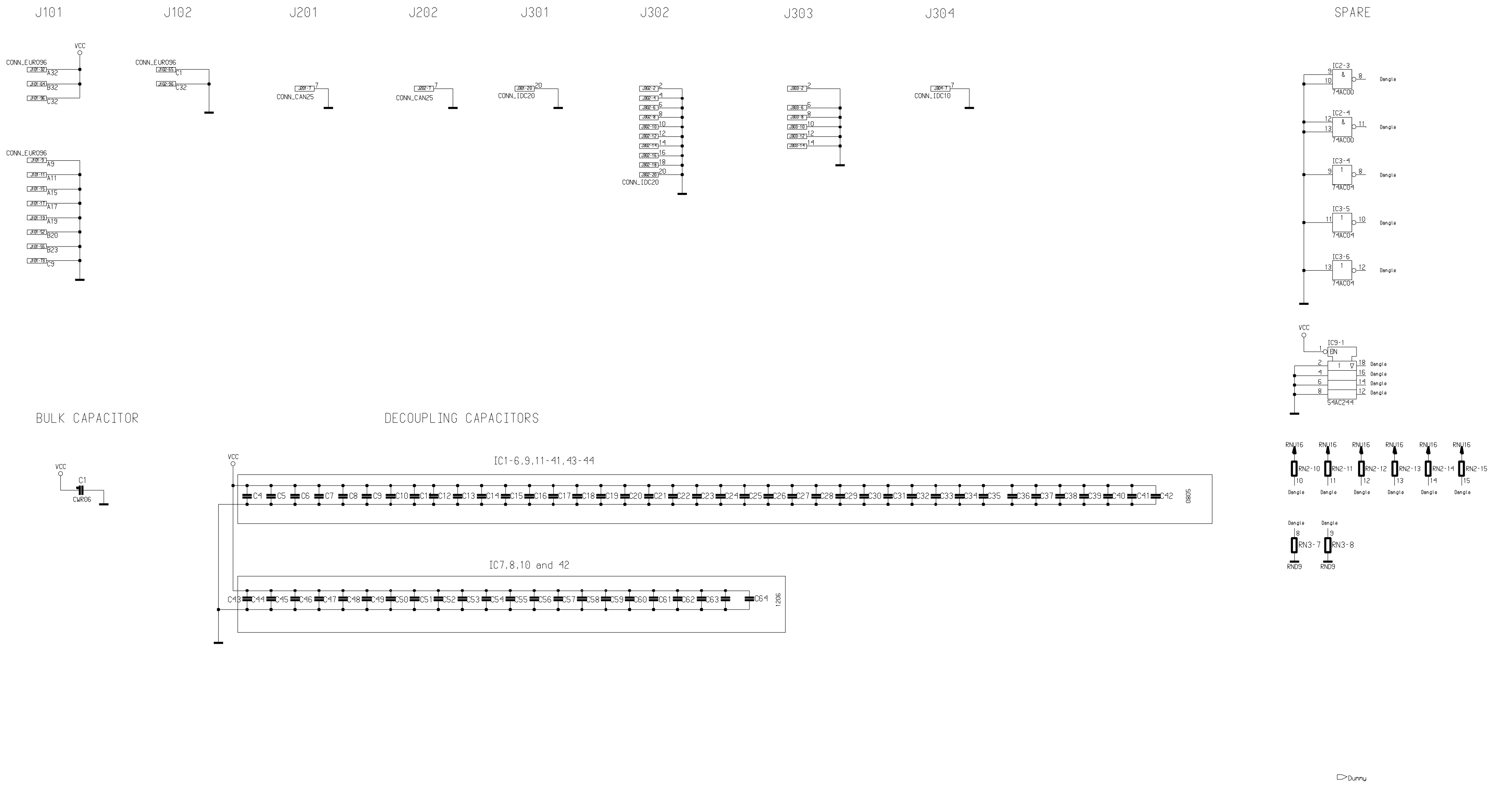
Saab Ericsson Space	Drwg No. 9241903-503	Issue A
	Designation DEM32 Io CIRCUIT DIAGRAM	Sheet No. 11/13

ORGINALFORMAT A1



Saab Ericsson Space	Drwg No. 9241903-503	Issue A
The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.	Designation CIRCUIT DIAGRAM DEM32 DMA	Sheet No. 12/13

ORGINALFORMAT A1



Saab Ericsson Space	Drwg No. 9241903-503	Issue A
The copyright/ownership of this document is and will remain ours. The document must not be used without our authorization or brought to the knowledge of a third party. Contravention will be prosecuted.	Designation CIRCUIT DIAGRAM DEM32 Power	Sheet No. 13/13