

AN204 Using nvSRAM in drives

Introduction

Electrical drives may have to move heavy loads in environments with increased safety requirements (humans are to be protected from accidents, malfunction cannot be allowed). To respond to these requirements drives are small industrial computers with communication interfaces, local memory and high power interfaces which are able to smoothly control powerful electric motors.

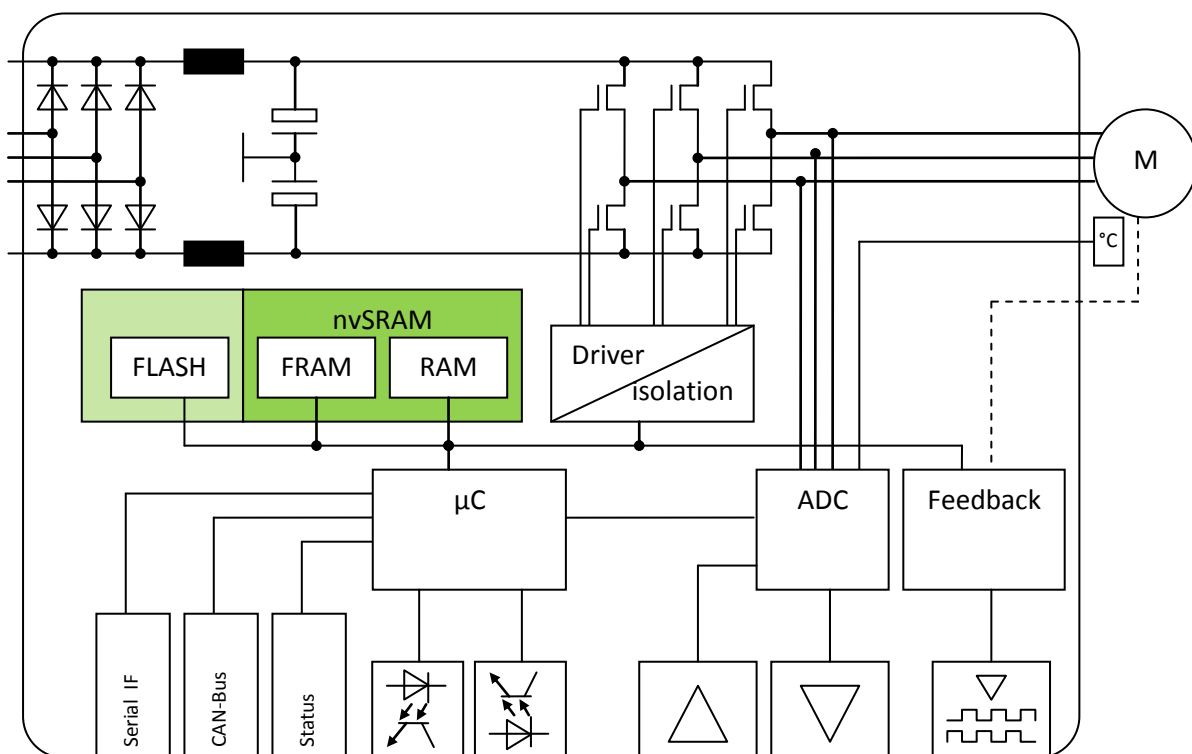


Fig.1 Block Diagram Driver

Drives safely start and slow down motors based on profiles which may be reprogrammed in the field. They monitor motor health and log wear and deviations from normal operation, as well as critical events and errors.

To fulfill these needs a designer needs nonvolatile memories with the following features:

- excellent data retention
- fast and easy programming
- unlimited number of Read and Write cycles

Nonvolatile SRAM perfectly fits these requirements.

In addition Anvo-Systems Dresden has implemented safety features to prevent corrupt data access to memory, in noisy environments. Using SecureWrite, corrupt addresses and damaged data will be automatically detected before the transferred data are written to the internal memory array. Sensitive data will not be overwritten, data won't be lost and only incorrupt data will be written.

Using nvSRAM in Drive Applications

In Drive Applications one nvSRAM can replace up to three types of memory: FLASH, emerging memory and SRAM.

Since nvSRAMs are ordinary SRAMs with the ability for self-sufficient, automatic backup of SRAM-data in an internal FLASH, they can be easily used to store drive profiles, configurations and similar data, which are typically stored in a FLASH. There is no need to erase cells before writing to an nvSRAM. New data can directly overwrite existing data at the target address.

Anvo-Systems Dresden nvSRAMs write at bus speed. Once a write instruction has been sent to the device, the next random write instruction can follow, there is no need to wait for internal programming cycles to finish.

In order to grant data integrity, even in harsh environments, Anvo-Systems Dresden has introduced the option for secure RAM access. The idea is to calculate a checksum from all transferred address and data bits and append this checksum to the transferred data. The transferred checksum is then compared to an internally calculated checksum; if the comparison fails the transferred data will be refused.

Transferred Data			Check CRC	Resulting Action	
Address	Data	CRC		Secure READ	Secure WRITE
o.k.	o.k.	o.k.	o.k.	accept data	write data to array
fail	don't care	don't care	fail	discard data, read again	discard data, set error flag
don't care	fail	don't care	fail	discard data, read again	discard data, set error flag
don't care	don't care	fail	fail	discard data, read again	discard data, set error flag

Checksum calculation and comparison is executed by dedicated hardware. Except for the transfer of the checksum, there is no communication overhead. Using SecureWrite it is virtually impossible to damage sensitive data such as driver profiles.

State-of-the-art drives can obtain health-data of the connected motor. These data need to be logged permanently. The high write speed of nvSRAMs and the fact that they don't suffer from limited R/W endurance supports the efficient implementation of log files, circular buffers or even event data recorders. All this data remains available even after unexpected power loss.

STORE and RECALL

Anvo-Systems Dresden nvSRAMs combine two types of memory, fast SRAM and energy-efficient SONOS-FLASH, on one single chip. Data transfer between the two memory-arrays runs automatically and is extremely fast.

Moving the FLASH data to the SRAM is called RECALL and is executed at power up or on software instruction. RECALL is powered by system power (VCC) and takes only 50 µs.

Transfer of SRAM-Data to the SONOS-FLASH is called STORE and can be executed either on software instruction (SoftStore) or automatically triggered (AutoStore) in a brown out situation or at power loss. While SoftStore is powered by VCC, AutoStore is executed self-sufficiently by using an independent source of energy. This source of energy is a small capacitor connected to the VCAP-pin. The capacitor is charged at power up, maintained during normal operation and expends the energy for reliable AutoStore execution while the remaining system powers down.

STORE operations take around 8ms. This time is comparable to the time an EEPROM or FLASH needs to execute a WRITE operation.

Document History:

Revision	Date	Modifications
0.0	23.05.2012	Initial Version