AN1157
APPLICATION NOTE

Connecting the 80960JA Microcontroller to M29 Series Flash Memories

CONTENTS

INTRODUCTION

This application note describes a method to connect an M29W004B Flash memory to an 80960JA microcontroller. The application note can be used as a reference for other Flash memory devices from STMicroelectronics.

The M29W004B is a 4 Mbit (512kb x 8) Flash memory from ST-Microelectronics, with asymmetrical block sizes. Other Flash parts that can be used in place of the M29W004B include the M29W002B, M29W008A and M29W116B. The TSOP40 packages of these memories are pin-compatible with the M29W004B, allowing them to be used in place of an M29W004B. Designers should track their PCBs for the extra address lines of the larger parts so they can be accommodated, if necessary, in the future.

The 80960JA is a member of Intel's i960 family of embedded RISC microprocessors. The i960 family has a large range of uses from I/O processors to computer peripherals such as printers.

ADVANTAGES OF FLASH

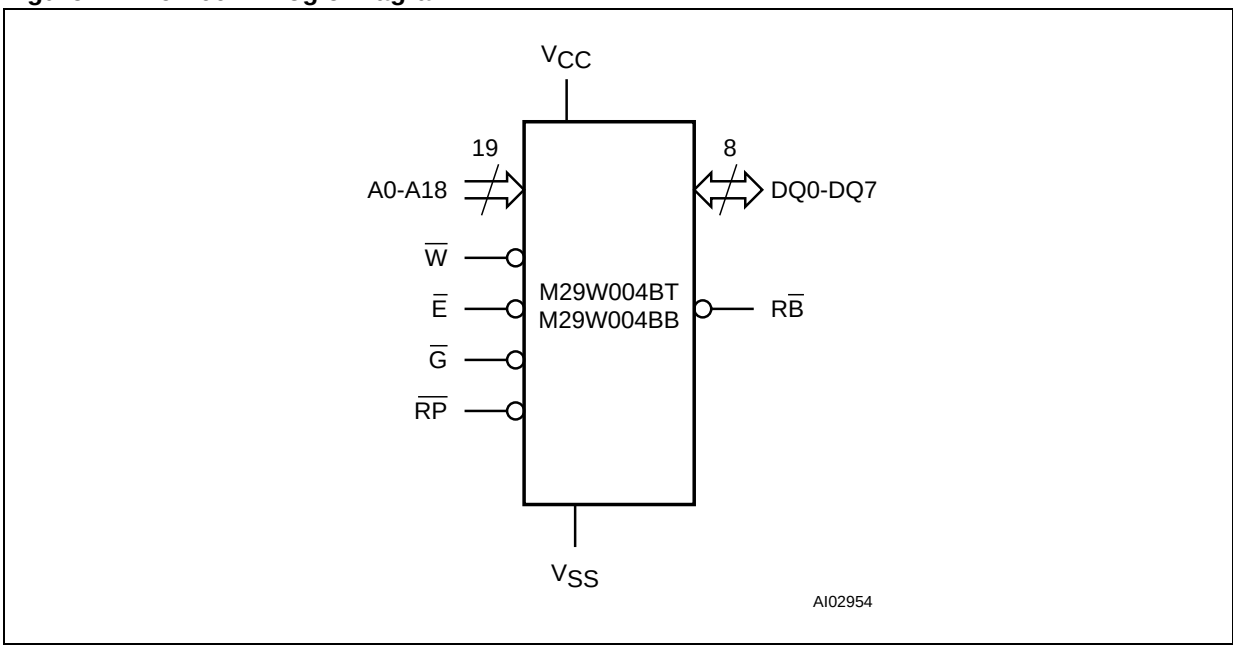
Flash memories can be used to store both code and data for the i960 microprocessor. Unlike EPROMs the data in Flash memories can be changed by the microprocessor. This enables non-volatile user data to be stored in the Flash. Field upgrades of the application code can be performed without any disassembly, unlike EPROM solutions.

It is usual to write separate boot and application programs so that the application program can be upgraded without changing the boot program. If the upgrade fails then the processor will still boot and it will be possible to reattempt to upgrade the application. The boot code should be programmed into the Flash before the Flash is fitted to the circuit board, otherwise it may not be possible to boot the microprocessor. Often the block containing the boot program is protected so it cannot become corrupt.

FLASH BUS ARCHITECTURE

Take a look at the bus on the M29W004B, Figure 1 shows the Logic Diagram.

Figure 1. M29W004B Logic Diagram



The memory has separate Address and Data Buses, the multiplexed bus of the 80960JA will need to be decoded and latched in order to connect correctly. The control lines are Chip Enable ($\overline{E}$), Output Enable ($\overline{G}$) and Write Enable ($\overline{W}$), these too will require additional logic.

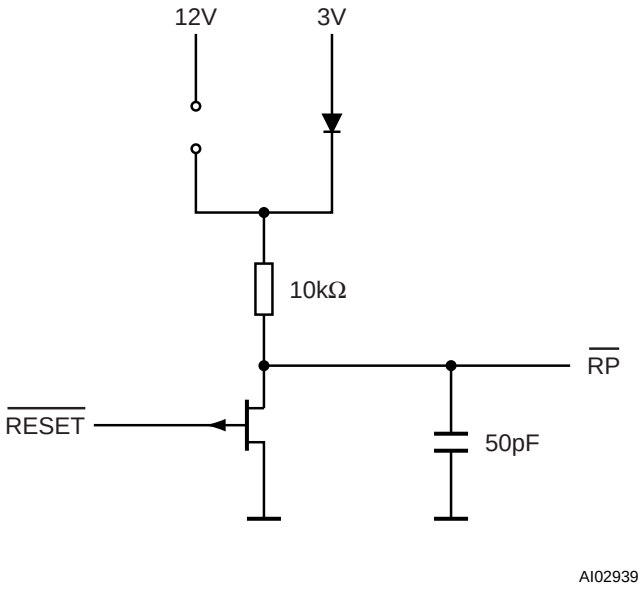
The Reset/Block Temporary Unprotect pin ($\overline{RP}$) accepts three states: Reset (V_{IL}), Not Reset (V_{IH}) and Block Temporary Unprotect (V_{ID}). Reset and Not Reset are the usual signals for a System Reset line. The third state, Block Temporary Unprotect is used to temporarily unprotect blocks that have been specifically protected in the memory. Many applications do not protect any blocks and therefore connect the $\overline{RP}$ pin directly to the system $\overline{RESET}$ signal.

Figure 2 gives an example of how the connection between the system $\overline{RESET}$ line and the M29W004B's $\overline{RP}$ pin can be made. The circuit makes use of a jumper to enable Block Temporary Unprotect. Many applications will provide the 12V from an external source, in which case the jumper can be replaced by a connector. The advantage with the circuit, as it stands, is that the system $\overline{RESET}$ will override Block Temporary Unprotect and cause the Flash to reset. Only four additional components are required.

Before the jumper is inserted, and when $\overline{RESET}$ is High, V_{IH}, $\overline{RP}$ is connected to 3V through the 10kΩ resistor and the diode. The current required by $\overline{RP}$ is very low, in the order of 1µA at 3V. The voltage drop in the resistor and the diode at these currents will keep $\overline{RP}$ very close to 3V. When the jumper is fitted the diode ceases to conduct and $\overline{RP}$ rises to 12V as the capacitor charges. The time-constant of a 10kΩ resistor and a 50pF capacitor is 500ns, satisfying the t_{PHPHH} rise-time requirements of the M29W004B. During a Reset, $\overline{RESET}$ is Low, V_{IL}, and the JFET is switched on, bringing $\overline{RP}$ close to ground. The current consumption during a Reset rises due to the current through the 10kΩ resistor.

Although the use of a jumper may not be the most elegant solution, it is a practical one because it maintains the security level offered by Block Protection. There is little point in having the Block Temporary Unprotect pin under software control. The whole point of the Block Protection feature is to protect against software failure. Allowing the Block Temporary Unprotect feature to be under the control of software is nearly equivalent to not protecting the blocks in the first place.

Figure 2. Reset/Block Temporary Unprotect Circuit



80960JA BUS ARCHITECTURE

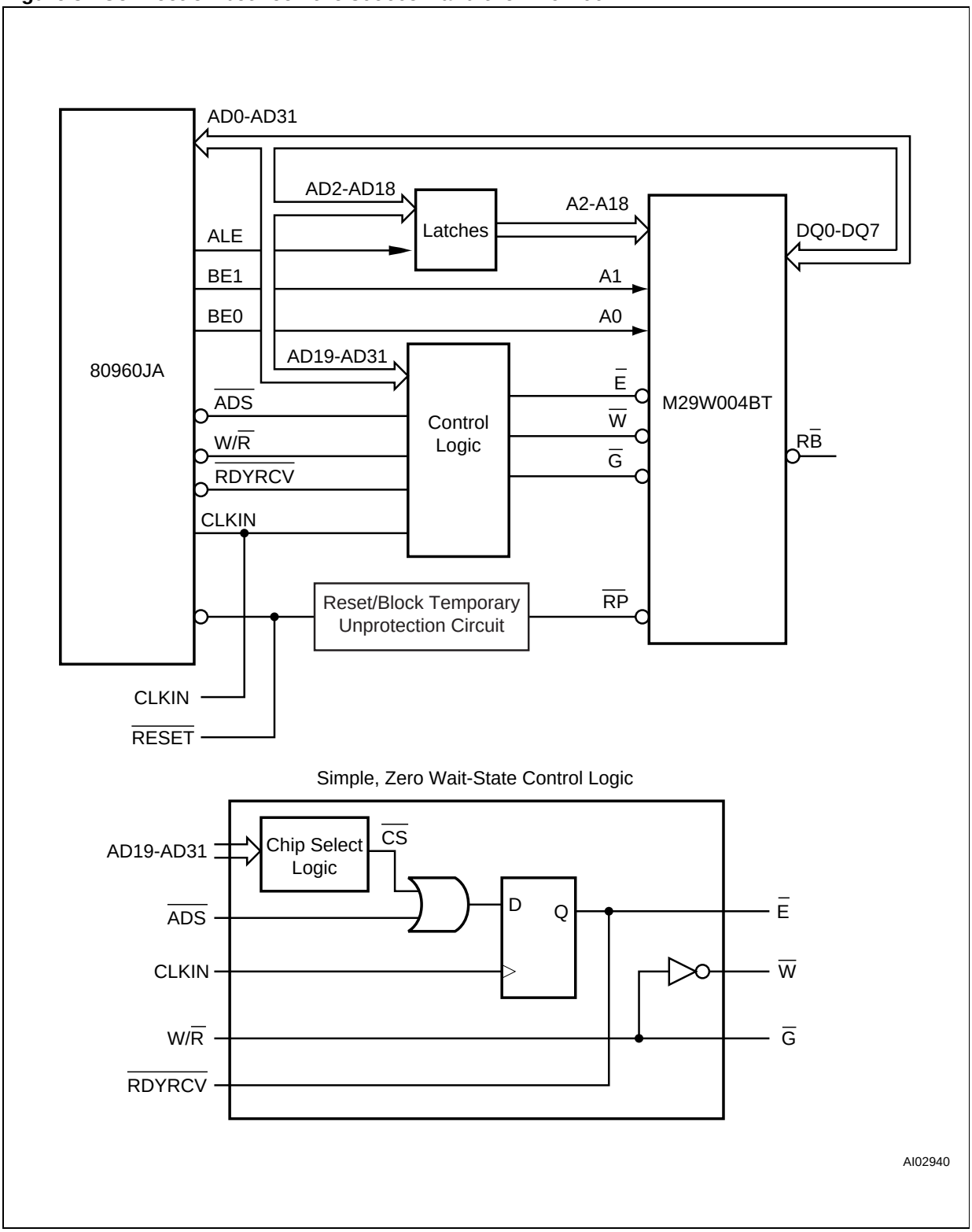
The 80960JA's bus architecture is very different from the architecture of the M29W004B. It is a synchronous interface, with a multiplexed bus and burst mode capability. There are many different control lines, none of which interface directly with the Flash. The bus supports 8-bit, 16-bit and 32-bit transfers, it supports software-transparent reading and writing of unaligned data.

Three of the five basic bus states are of interest to a design interfacing an M29W004B to the 80960JA, the T_a state (address), $T_{w/d}$ (wait/data) and T_r (recovery). During the T_a state the address is output on the multiplexed bus, during the $T_{w/d}$ state the data is read or written to the bus and during the T_r state the memory has time to release the bus.

The address needs to be latched during the T_a state; the Flash memory requires the address to be valid during the entire read or write cycle. The 80960JA includes an output, ALE (address latch enable), which can be used to latch the address at the correct time. The address requires decoding to a Chip Select signal, which is low when the microprocessor addresses the Flash. During the $T_{w/d}$ state the R/$\overline{W}$ signal requires decoding to Output Enable and Write Enable signals; a wait-state generator is required to signal the microprocessor that the Flash is ready. Finally, during the T_r state the Flash's Chip Enable must be brought high for the Flash to release the bus. The accesses required can be encoded into a state machine (best when one or more wait-states are required). For zero wait-states a simpler circuit can be devised, Figure 3 shows a simple connection.

The Chip Select Logic works directly off the multiplexed address/data bus. By ORing the $\overline{CS}$ with $\overline{ADS}$ the input of the D-type only clocks a valid Chip Select when a valid Flash address is on AD19-AD31. The Chip Select is latched to provide the Flash with the Chip Enable, $\overline{E}$, signal. The Output Enable signal for the Flash is taken directly from the W/$\overline{R}$ signal, and inverted to give the Write Enable signal. The $\overline{RDYRCV}$ signal that tells the microprocessor that the Flash is ready is generated directly from the Chip Enable signal, since the 80960JA is a synchronous bus this signal is only sampled when the clock rises, and therefore Chip Enable generates the correct waveform.

Figure 3. Connection between the 80960JA and the M29W004B



In order to boot the 80960JA from the code in the M29W004B it is necessary to map the M29W004B to the address space where the 80960JA holds its Initialization Boot Record (FEFF FF30h-FEFF FF5F). The 80960JA reads the Initialization Boot Record using 8-bit accesses, making an 8-bit Flash part very suitable for booting the 80960JA.

Since the 80960JA looks at the top of its external memory space for the Initialization Boot Record the usual choice of Flash is a top-boot block part, such as the M29W004BT. The M29W004BT can be mapped to have its boot block at the address of the Initialization Boot Record, all the initialization data structures and the boot code can be put in the boot-block of the M29W004BT.

Typically the application software would not be run directly from an 8-bit Flash. 8-bit Flash is still an excellent storage media for application software since it has a fast access time per byte, it is simple to use and is relatively inexpensive compared to 16-bit and 32-bit Flash. The boot code should copy the application code to a 32-bit wide memory (such as DRAM) and run it from there. There are Flash memories that are suitable for running the application code directly from the Flash; they include burst/page mode bus interfaces to minimize the access times and dual-bank architectures so the program can be run from Flash while the Flash is erasing or programming. A 16-bit or 32-bit wide Flash should be used with the 80960JA processor if the program is to be run directly from it.

TIMING REQUIREMENTS

Each of the bus states for the 80960JA lasts one clock period, for a 16.67MHz clock the access time will be 60ns less the delay in the D-type flip-flop. The following timings have been derived for a flip-flop with a 5ns delay. Table 1 and Figure 4 shown the Read Timings; Table 2 and Figure 5 show the Write Timings.

Table 1. Principal Read Timing Requirements

M29W004B				80960JA
Symbol	**55**	**70**	**90**	**5ns flip-flop**
t_{AVAV}	55	70	90	60
t_{AVQV}	55	70	90	60
t_{ELQV}	55	70	90	55
t_{GLQV}	30	30	35	96.5
t_{EHQZ}	20	25	30	55
t_{OH}	0	0	0	0

Figure 4. Principal Read Timing Waveform

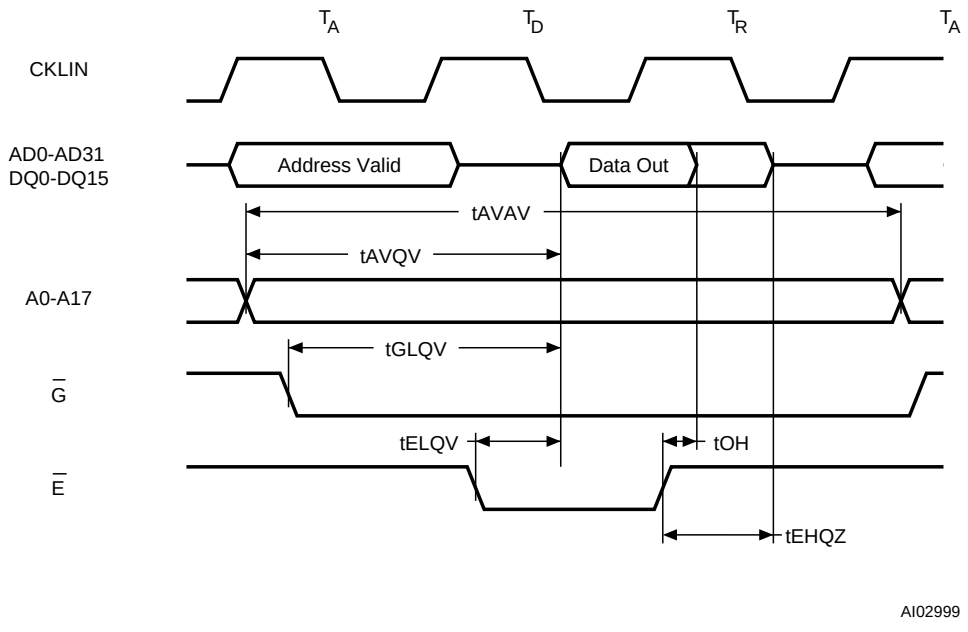
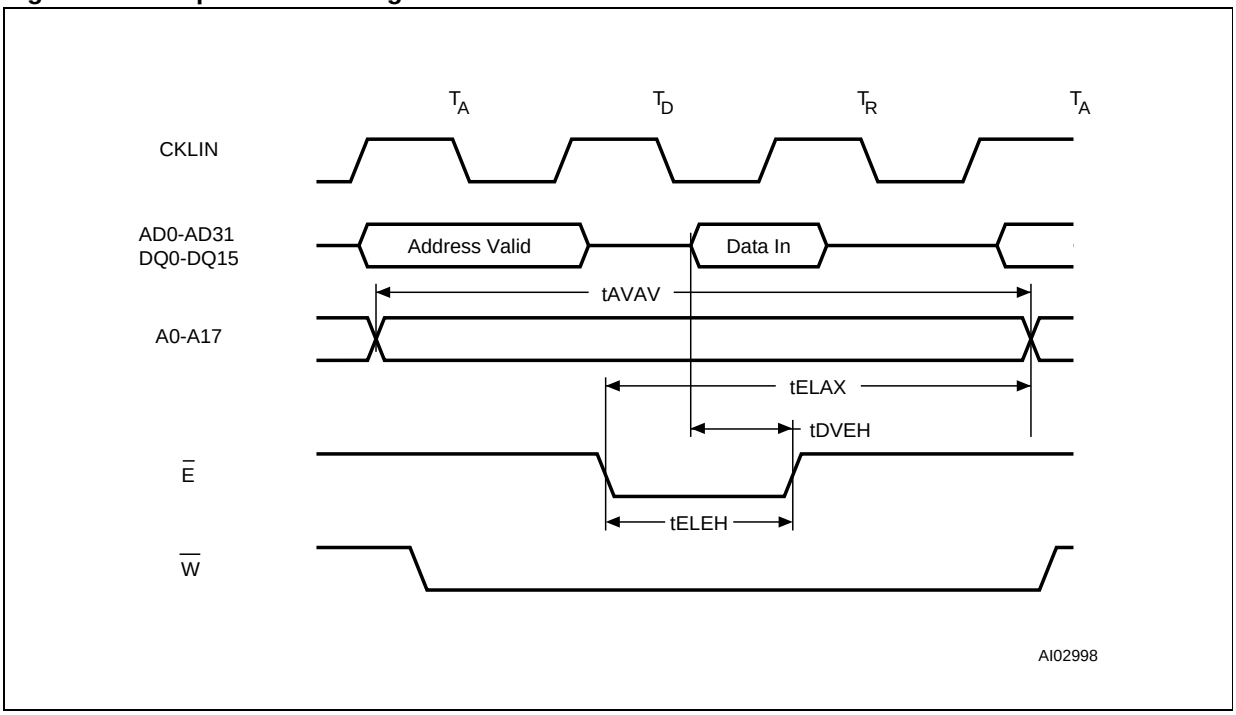


Table 2. Principal Write Timing Requirements, Write Enable Controlled

M29W004B				80960JA
Symbol	**55**	**70**	**90**	**5ns flip-flop**
t_{AVAV}	55	70	90	60
t_{ELEH}	40	45	45	60
t_{DVEH}	25	30	45	48.5
t_{EHEL}	30	30	30	120
t_{ELAX}	40	45	45	115

Figure 5. Principal Write Timing Waveforms



From the timings it can be seen that a 55ns Flash is required to provide zero wait-state access with a 5ns flip-flop. For slower parts it will be necessary to implement a state-machine to add additional wait-states.

CONCLUSION

The M29W004A and other STMicroelectronics Flash can be connected to the 80960JA as the boot ROM with zero wait-state access. The M29W004B provides embedded systems with non-volatile data storage and the ability to perform field upgrades of application software without disassembly.

If you have any questions or suggestion concerning the matters raised in this document please send them to the following electronic mail address:

ask.memory@st.com (for general enquiries)

Please remember to include your name, company, location, telephone number and fax number.