**Welcome to E-XFL.COM**

## Understanding Embedded - Microprocessors

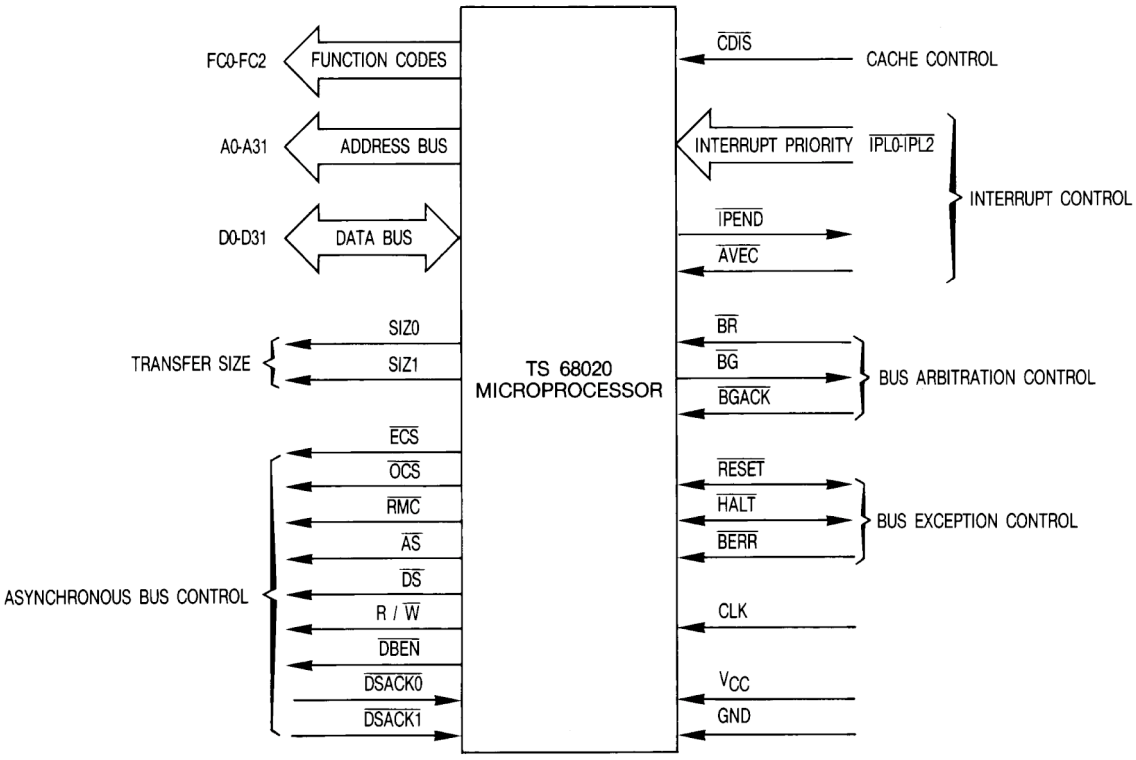
Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

## Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | 68000 |
| Number of Cores/Bus Width | 1 Core, 32-Bit |
| Speed | 16.67MHz |
| Co-Processors/DSP | - |
| RAM Controllers | - |
| Graphics Acceleration | No |
| Display & Interface Controllers | - |
| Ethernet | - |
| SATA | - |
| USB | - |
| Voltage - I/O | 5.0V |
| Operating Temperature | -55°C ~ 125°C (TC) |
| Security Features | - |
| Package / Case | 114-BCPGA |
| Supplier Device Package | 114-CPGA (34.54x34.54) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/ts68020mr16 |

**Figure 4.** Functional Signal Groups




## Signal Description

Figure 4 illustrates the functional signal groups and Table 1 lists the signals and their function.

The $V_{CC}$ and GND pins are separated into four groups to provide individual power supply connections for the address bus buffers, data bus buffers, and all other output buffers and internal logic.

| Group | $V_{CC}$ | GND |
|---|---|---|
| Address Bus | A9, D3 | A10, B9,C3, F12 |
| Data Bus | M8, N8, N13 | L7, L11, N7, K3 |
| Logic | D1, D2, E3, G11, G13 | G12, H13, J3, K1 |
| Clock | — | B1 |

## Electrical Characteristics

**Table 2.** Absolute Maximum Ratings

| Symbol | Parameter | Test Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{CC}$ | Supply Voltage | | -0.3 | +7.0 | V |
| $V_I$ | Input Voltage | | -0.5 | +7.0 | V |
| $P_{dmax}$ | Max Power Dissipation | $T_{case}$ = -55°C | | 2.0 | W |
| | | $T_{case}$ = +125°C | | 1.9 | W |
| $T_{case}$ | Operating Temperature | M Suffix | -55 | +125 | °C |
| | | V Suffix | -40 | +85 | °C |
| $T_{stg}$ | Storage Temperature | | -55 | +150 | °C |
| $T_{leads}$ | Lead Temperature | Max 5 Sec. Soldering | | +270 | °C |

**Table 3.** Recommended Condition of Use

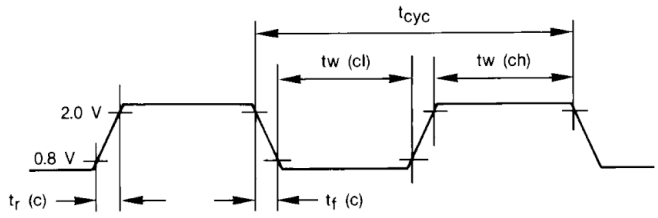
Unless otherwise stated, all voltages are referenced to the reference terminal (see Table 1).

| Symbol | Parameter | | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{CC}$ | Supply Voltage | | 4.5 | 5.5 | V |
| $V_{IL}$ | Low Level Input Voltage | | -0.3 | 0.5 | V |
| $V_{IH}$ | High Level Input Voltage | | 2.4 | 5.25 | V |
| $T_{case}$ | Operating Temperature | | -55 | +125 | °C |
| $R_L$ | Value of Output Load Resistance | | (1) | | Ω |
| $C_L$ | Output Loading Capacitance | | | (1) | pF |
| $t_r(c)$–$t_f(c)$ | Clock Rise Time (See Figure 5) | 68020-16 | | 5 | ns |
| | | 68020-20 | | 5 | |
| | | 68020-25 | | 4 | |
| $f_c$ | Clock Frequency (See Figure 5) | 68020-16 | 8 | 16.67 | MHz |
| | | 68020-20 | 12.5 | 20 | |
| | | 68020-25 | 12.5 | 25 | |
| $t_{cyc}$ | Cycle Time (see Figure 5) | 68020-16 | 60 | 125 | ns |
| | | 68020-20 | 50 | 80 | |
| | | 68020-25 | 40 | 80 | |
| $t_W(CL)$ | Clock Pulse Width Low (See Figure 5) | 68020-16 | 24 | 95 | ns |
| | | 68020-20 | 20 | 54 | |
| | | 68020-25 | 19 | 61 | |
| $t_W(CH)$ | Clock Pulse Width High (See Figure 5) | 68020-16 | 24 | 95 | ns |
| | | 68020-20 | 20 | 50 | |
| | | 68020-25 | 19 | 61 | |

Note: 1. Load network number 1 to 4 as specified (Table 7) gives the maximum loading of the relevant output.

This device contains protective circuitry against damage due to high static voltages or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or $V_{CC}$).

**Figure 5.** Clock Input Timing Diagram




Note: Timing measurements are referenced to and from a low voltage of 0.8V and a high voltage of 2.0V, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8V and 2.0V.

**Table 4.** Thermal Characteristics at 25°C

| Package | Symbol | Parameter | Value | Unit |
|---|---|---|---|---|
| PGA 114 | $\theta_{JA}$ | Thermal Resistance - Ceramic Junction to Ambient | 26 | °C/W |
| | $\theta_{JC}$ | Thermal Resistance - Ceramic Junction to Case | 5 | °C/W |
| CQFP 132 | $\theta_{JA}$ | Thermal Resistance - Ceramic Junction to Ambient | 34 | °C/W |
| | $\theta_{JC}$ | Thermal Resistance - Ceramic Junction to Case | 2 | °C/W |

## Power Considerations

The average chip-junction temperature, $T_J$, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

$T_A$ = Ambient Temperature, °C

$\theta_{JA}$ = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{CC} \cdot V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between $P_D$ and $T_J$ (if $P_{I/O}$ is neglected) is:

$$P_D = K + (T_J + 273) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part K can be determined from equation **(3)** by measuring $P_D$ (at equilibrium) for a known $T_A$. Using this value of K, the values of $P_D$ and $T_J$ can be obtained by solving equations **(1)** and **(2)** iterativley for any value of $T_A$.

## Dynamic (Switching) Characteristics

The limits and values given in this section apply over the full case temperature range -55°C to +125°C and $V_{CC}$ in the range 4.5V to 5.5V $V_{IL}$ = 0.5V and $V_{IH}$ = 2.4V (See also note 12 and 13). The INTERVAL numbers refer to the timing diagrams. See Figure 5, Figure 9 and Figure 12.

**Table 6.** Dynamic Electrical Characteristics

| Symbol | Parameter | Interval Number | 68020-16 | | 68020-20 | | 68020-25 | | Unit | Notes |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | | |
| $t_{CPW}$ | Clock Pulse Width | 2 , 3 | 24 | 95 | 20 | 54 | 19 | 61 | ns | |
| $t_{CHAV}$ | Clock High to Address/FC/Size/$\overline{RMC}$ Valid | 6 | 0 | 30 | 0 | 25 | 0 | 25 | ns | |
| $t_{CHEV}$ | Clock High to $\overline{ECS}$, $\overline{OCS}$ Asserted | 6A | 0 | 20 | 0 | 15 | 0 | 12 | ns | |
| $t_{CHAZX}$ | Clock High to Address/Data/FC/$\overline{RMC}$/ Size High Impedance | 7 | 0 | 60 | 0 | 50 | 0 | 40 | ns | (11) |
| $t_{CHAZn}$ | Clock High to Address/FC/Size/$\overline{RMC}$ Invalid | 8 | 0 | | 0 | | 0 | | ns | |
| $t_{CLSA}$ | Clock Low to $\overline{AS}$, $\overline{DS}$ Asserted | 9 | 3 | 30 | 3 | 25 | 3 | 18 | ns | |
| $t_{STSA}$ | $\overline{AS}$ to $\overline{DS}$ Assertion (Read)(Skew) | 9A | -15 | 15 | -10 | 10 | -10 | 10 | ns | (1) |
| $t_{ECSA}$ | $\overline{ECS}$ Width Asserted | 10 | 20 | | 15 | | 15 | | ns | |
| $t_{OCSA}$ | $\overline{OCS}$ Width Asserted | 10A | 20 | | 15 | | 15 | | ns | |
| $t_{EOCSN}$ | $\overline{ECS}$, $\overline{OCS}$ Width Negated | 10B | 15 | | 10 | | 5 | | ns | (11) |
| $t_{AVSA}$ | Address/FC/Size/$\overline{RMC}$ Valid to $\overline{AS}$ Asserted (and $\overline{DS}$ Asserted, Read) | 11 | 15 | | 10 | | 6 | | ns | (6) |
| $t_{CLSN}$ | Clock Low to $\overline{AS}$, $\overline{DS}$ Negated | 12 | 0 | 30 | 0 | 25 | 0 | 15 | ns | |
| $t_{CLEN}$ | Clock Low to $\overline{ECS}$/$\overline{OCS}$ Negated | 12A | 0 | 30 | 0 | 25 | 0 | 15 | ns | |
| $t_{SNAI}$ | $\overline{AS}$, $\overline{DS}$ Negated to Address/FC/ Size/$\overline{RMC}$ Invalid | 13 | 15 | | 10 | | 10 | | ns | |
| $t_{SWA}$ | $\overline{AS}$ (and $\overline{DS}$, Read) Width Asserted | 14 | 100 | | 85 | | 70 | | ns | |
| $t_{SWAW}$ | $\overline{DS}$ Width Asserted, Write | 14A | 40 | | 38 | | 30 | | ns | |
| $t_{SN}$ | $\overline{AS}$, $\overline{DS}$ Width Negated | 15 | 40 | | 38 | | 30 | | ns | (11) |
| $t_{SNSA}$ | $\overline{DS}$ Negated to $\overline{AS}$ Asserted | 15A | 35 | | 30 | | 25 | | ns | (8) |
| $t_{CSZ}$ | Clock High to $\overline{AS}$/$\overline{DS}$/R/$\overline{W}$/$\overline{DBEN}$ High Impedance | 16 | | 60 | | 50 | | 40 | ns | (11) |
| $t_{SNRN}$ | $\overline{AS}$, $\overline{DS}$ Negated to R/$\overline{W}$ High | 17 | 15 | | 10 | | 10 | | ns | (6) |
| $t_{CHRH}$ | Clock High to R/$\overline{W}$ High | 18 | 0 | 30 | 0 | 25 | 0 | 20 | ns | |
| $t_{CHRL}$ | Clock High to R/$\overline{W}$ Low | 20 | 0 | 30 | 0 | 25 | 0 | 20 | ns | |
| $t_{RAAA}$ | R/$\overline{W}$ High to $\overline{AS}$ Asserted | 21 | 15 | | 10 | | 5 | | ns | (6) |
| $t_{RASA}$ | R/$\overline{W}$ Low to $\overline{DS}$ Asserted (Write) | 22 | 75 | | 60 | | 50 | | ns | (6) |
| $t_{CHDO}$ | Clock High to Data Out Valid | 23 | | 30 | | 25 | | 25 | ns | |
| $t_{SNDI}$ | AS, DS Negated to Data Out Valid | 25 | 15 | | 10 | | 5 | | ns | (6) |
| $t_{DNDBN}$ | $\overline{DS}$ Negated to $\overline{DBEN}$ Negated (Write) | 25A | 15 | | 10 | | 5 | | ns | (9) |

## Input and Output Signals for Dynamic Measurements

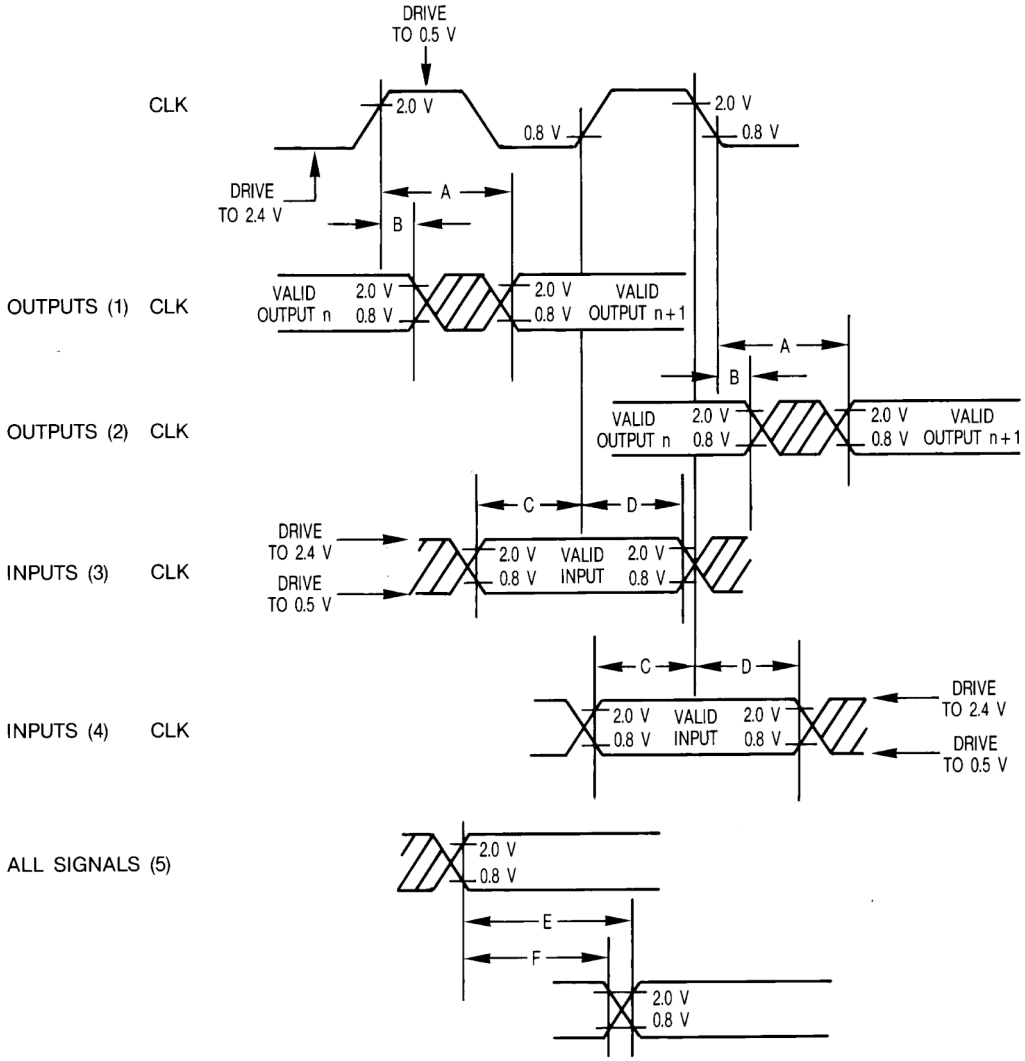
### AC Electrical Specifications Definitions

The AC specifications presented consist of output delays, input setup and hold times, and signal skew times. All signals are specified relative to an appropriate edge of the TS68020 clock input and, possibly, relative to one or more other signals.

The measurement of the AC specifications is defined by the waveforms in Figure 12. In order to test the parameters guaranteed by Atmel, inputs must be driven to the voltage levels specified in Figure 12. Outputs of the TS68020 are specified with minimum and/or maximum limits, as appropriate, and are measured as shown. Inputs to the TS68020 are specified with minimum and, as appropriate, maximum setup and hold times, and are measurement as shown. Finally, the measurements for signal-to-signal specification are also shown.

Note that the testing levels used to verify conformance of the TS68020 to the AC specifications does not affect the guaranteed DC operation of the device as specified in the DC electrical characteristics.

**Figure 12.** Drive Levels and Test Points for AC Specification




Legend:

A) Maximum Output Delay Specification

B) Minimum Output Hold Time

C) Minimum Input Setup Time Specification

D) Minimum Input Hold Time Specification

E) Signal Valid to Signal Valid Specification (Maximum or Minimum)

F) Signal Valid to Signal Invalid Specification (Maximum or Minimum)

Notes:
1. This output timing is applicable to all parameters specified relative to the rising edge of the clock.
2. This out put timing is applicable to all parameters specified relative to the falling edge of the clock.
3. This input timing is applicable to all parameters specified relative to the falling edge of the clock.
4. This input timing is applicable to all parameters specified relative to the falling edge of the clock.
5. This timing is applicable to all parameters specified relative to the assertion/negation of another signal.

## Additional Information

Additional information shall not be for any inspection purposes.

### Power Consideration

See Table 4.

### Capacitance (Not for Inspection Purposes

| Symbol | Parameter | Test Conditions | Min | Unit |
|---|---|---|---|---|
| $C_{in}$ | Input Capacitance | $V_{in}$ = 0V $T_{amb}$ = 25°C<br>f = 1 MHz | 20 | pF |

## Capacitance Derating Curves

Figure 13 to Figure 18 inclusive show the typical derating conditions which apply. The capacitance includes any stray capacitance. The graphs may not be linear outside the range shown.

**Figure 13.** Address Capacitance Derating Curve

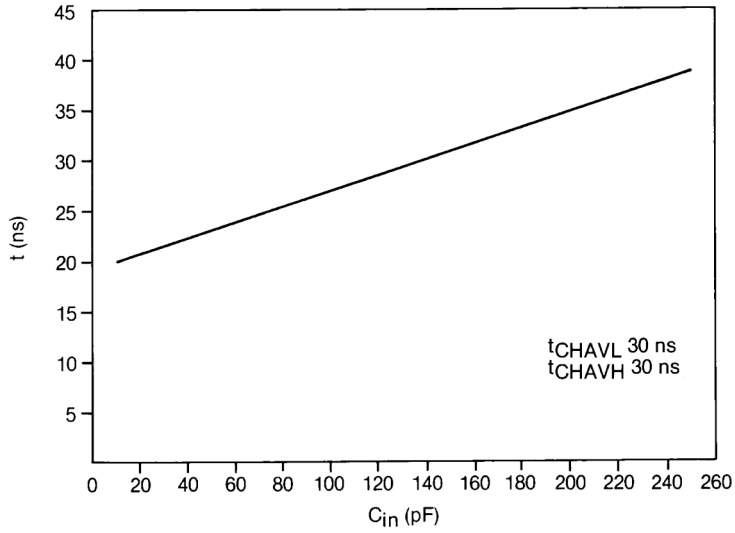


**Figure 16.** DS, AS, IPEND, and BG Capacitance Derating Curve

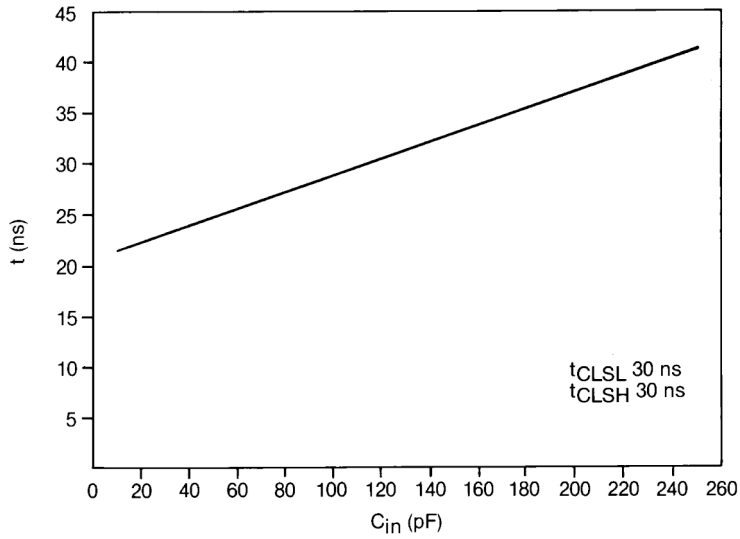
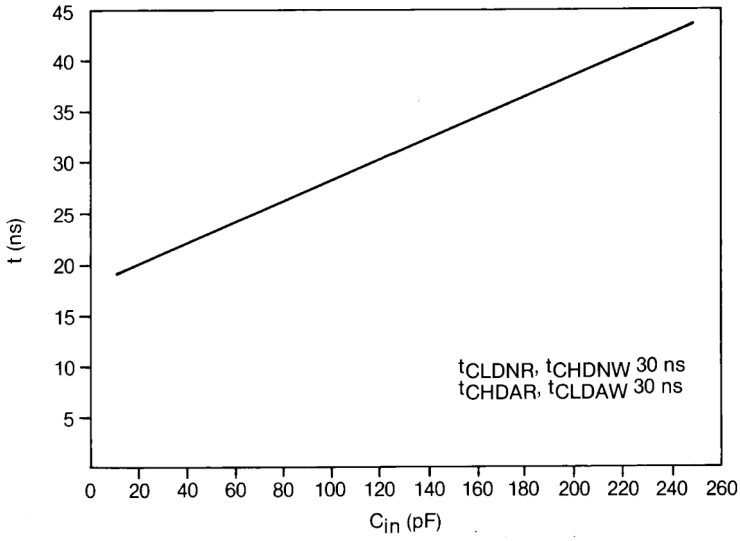



**Figure 17.** DBEN Capacitance Derating Curve



The TS68000 Family processors distinguish address spaces as supervisor / used and program/data. These four combinations are specified by the function code pins (FC0/FC1/FC2) during bus cycles, indication the particular address space. Using the function codes, the memory sub-system can distinguish between authorized access (supervisor mode is privileged access) and unauthorized access (user mode may not have access to supervisor program or data areas). To support the full privileges of the supervisor, the alternate function code registers allow the supervisor to specify an access to user program or data areas by preloading the SFC/DFC registers appropriately.

The cache registers (control — CACR, address — CAAR) allow software manipulation of the on-chip instruction cache. Control and status accesses to the instruction cache are provided by the cache control register (CACR), while the cache address register (CAAR) holds the address for those cache control functions that require an address.

**Figure 19.** User Programming Model

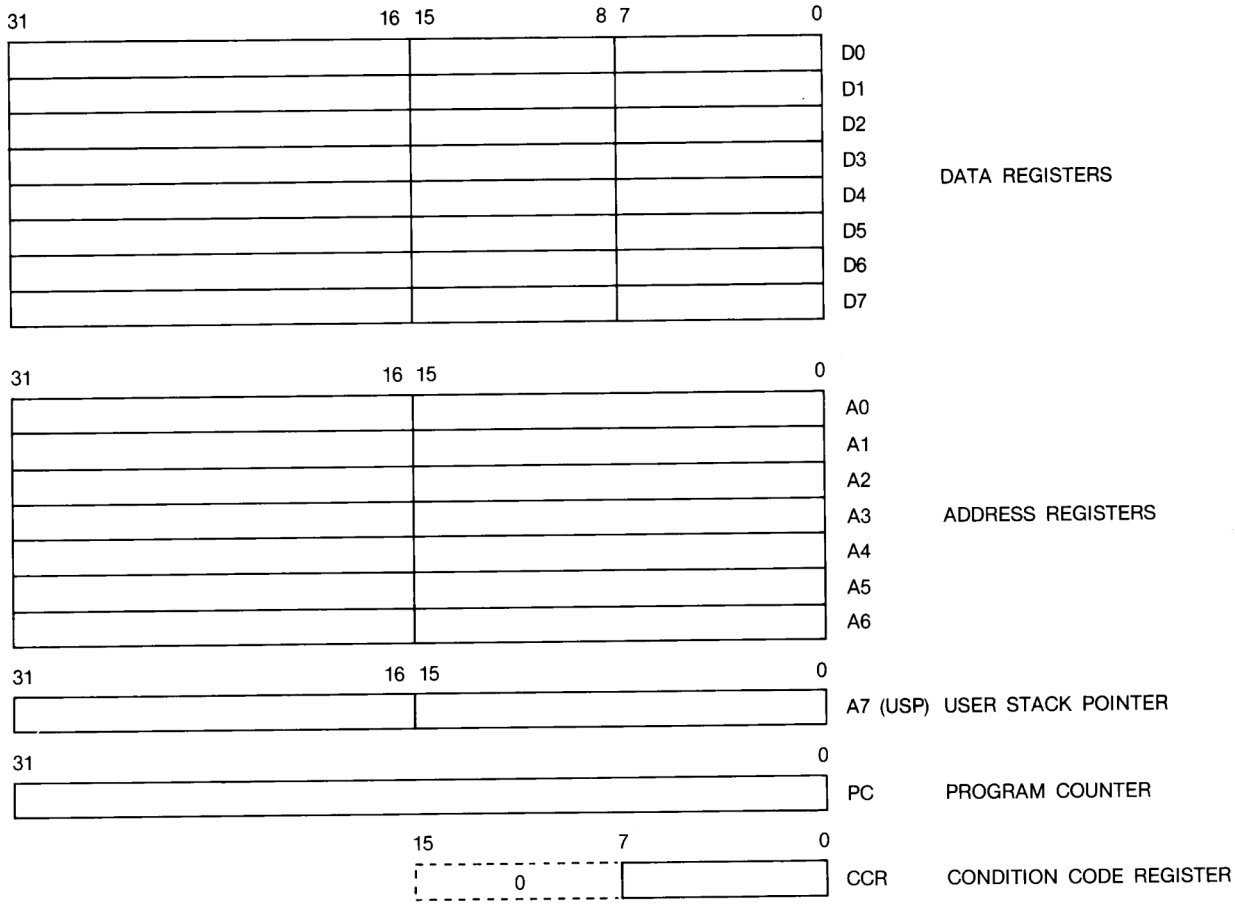


**Figure 20.** Supervisor Programming Model Supplement

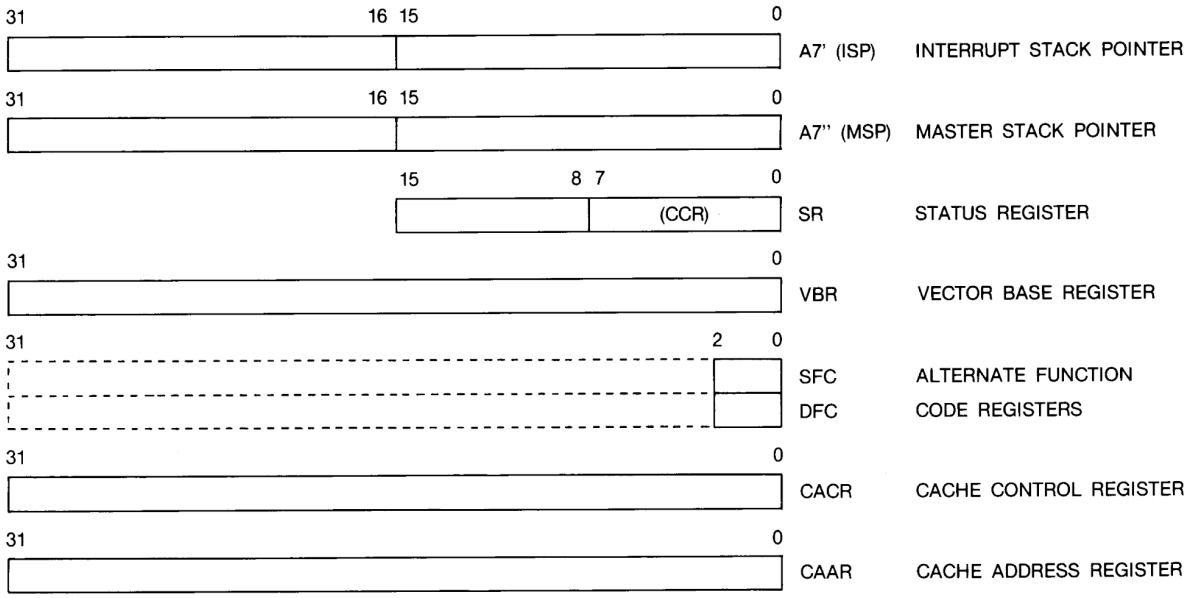
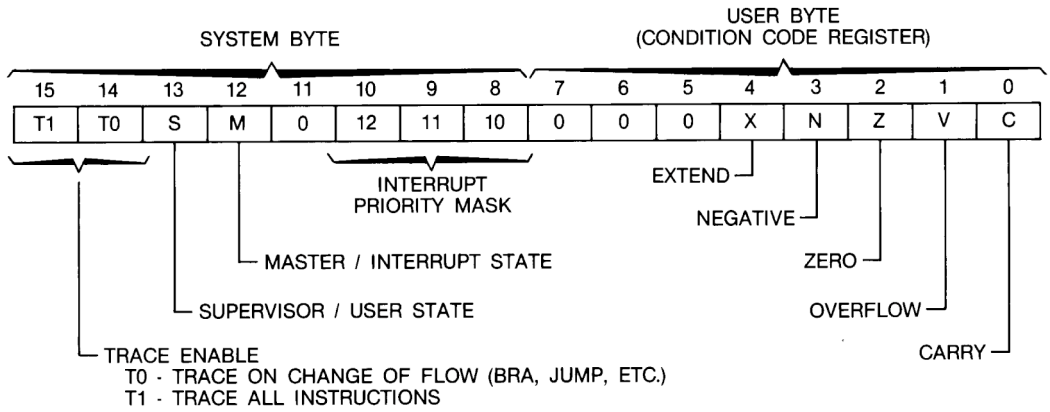



**Figure 21.** Status Register




## Data Types and Addressing Modes

Seven basic types are supported. These data types are:

- Bits
- Bits Flieds (String of consecutive bits, 1-32 bits long)
- BCD Digits (Packed: 2 digits/byte, Unpacked: 1 digit/byte)
- Byte Integers (8-bit)
- Word Integers (16-bit)
- Long Word Integers (32-bit)
- Quad Word Integers (64-bit)

In addition, operations on other data types, such as memory addresses, status word data, etc...., are provided in the instruction set. The co-processor mechanism allows direct support of floating-point data type with the TS68881 and TS68882 floating-point co-processors, as well as specialized user-defined data types and functions.

**Table 9.** Instruction Set (Continued)

| Mnemonic | Description |
|---|---|
| OR | Logical Inclusive OR |
| ORI | Logical Inclusive OR Immediate |
| PACK | Pack BCD |
| PEA | Push Effective Address |
| RESET | Reset External Devices |
| ROL, ROR | Rotate Left and Right |
| ROXL, ROXR | Rotate with Extend Left and Right |
| RTD | Return and Deallocate |
| RTE | Return and Exception |
| RTM | Return from Module |
| RTR | Return and Restore Codes |
| RTS | Return from Subroutine |
| SBCD | Subtract Decimal with Extend |
| Scc | Set Conditionally |
| STOP | Stop |
| SUB | Subtract |
| SUBA | Subtract Address |
| SUBI | Subtract Immediate |
| SUBQ | Subtract Quick |
| SUBX | Subtract with Extend |
| SWAP | Swap Register Words |
| TAS | Test Operand and Set |
| TRAP | Trap |
| TRAPcc | Trap Conditionally |
| TRAPV | Trap on Overflow |
| TST | Test Operand |
| UNLK | Unlink |
| UNPK | Unpack BCD |
| **Co-processor Instructions** | |
| cpBCC | Branch Conditionally |
| cpDBcc | Test Co-processor Condition, Decrement and Branch |
| cpGEN | Co-processor General Instruction |
| cpRESTORE | Restore Internal State of Co-processor |
| cpSAVE | Save Internal State of Co-processor |
| cpScc | Set Conditionally |
| cpTRAPcc | Trap Conditionally |

### Multi-processing

To further support multi-processing with the TS68020, a compare and swap instruction, CAS, has been added. This instruction makes use of the read-modify-write cycle to compare two operands and swap a third operand pending the results of the compare. A variant of this instruction, CAS2, performs similarly comparing dual operand pairs, and updating two operands.

These multi-processing operations are useful when using common memory to share or pass data between multiple processing elements. The read-modify-write cycle is an indivisible operand that allows reading and updating a “lock” operand used to control access to the common memory elements. The CAS2 instruction is more powerful since dual operands allow the “lock” to the checked and two values (i.e., both pointers in a doubly-linked list) to be updated according to the lock’s status, all in a single operation.

### Module Support

The TS68020 includes support for modules with the call module (CALLM) and return from module (RTM) instructions. The CALLM instruction references a module descriptor. This descriptor contains control information for entry into the associated module. The CALLM instruction creates a module stack frame and stores the module state in that frame. The RTM instruction recovers the previous module state from the stack frame and returns to the calling module.

The module interface also provides a mechanism for finer resolution of access control by external hardware. Although the TS68020 does not interrupt the access control information, it does communicate with external hardware when the access control is to be changed, and relies on the external hardware to verify that the changes are legal.

CALLM and RTM, when used as subroutine calls and returns with proper descriptor formats, cause the TS68020 to perform the necessary actions to verify legitimate access to modules.

## Virtual Memory/Machine Concepts

The full addressing range of the TS68020 is 4-Gbyte (4, 294, 967, 296). However, most TS68020 systems implement a smaller physical memory. Nonetheless, by using virtual memory techniques, the system can be made to appear to have a full 4-Gbyte of physical memory available to each user program. These techniques have been used for many years in large mainframe computers and minicomputers. With the TS68020 (as with the TS68010 and TS68012), virtual memory can be fully supported in microprocessor-based systems.

In a virtual memory system, a user program can be written as though it has a large amount of memory available to it when actually only a smaller amount of memory is physically present in the system. In a similar fashion, a system provides user programs access to other devices that are not physically present in the system, such as tape drives, disk drives, printers, or terminals. With proper software emulation, a physical system can be made to appear to a user program as any other 68000 computer system and the program may be given full access to all of the resources of that emulated system. Such an emulator system is called a virtual machine.

### Virtual Memory

The basic mechanism for supporting virtual memory is to provides a limited amount of high-speed physical memory that can be accessed directly by the processor while maintaining of a much larger “virtual” memory on secondary storage devices such as large capacity disk drives. When the processor attempts to access a location in the virtual memory map that is not resident in the physical memory (referred to as a page fault), the access to that location is temporarily suspended while the necessary data is fetched from secondary storage and placed in physical memory; the suspended access is then either restarted or continued.

The TS68020 will always transfer the maximum amount of data on all bus cycles; i.e., it always assumes the port is 32-bit wide when beginning the bus cycle. In addition, the TS68020 has no restrictions concerning alignment of operands in memory; long word operands need not be aligned on long word address boundaries. When misaligned data requires multiple bus cycles, the TS68020 aligned data requires multiple bus cycles, the TS68020 automatically runs the minimum number of bus cycles.

## The Co-processor Concept

The co-processor interface is a mechanism for extending the instruction set of the TS68000 Family. Examples of these extensions are the addition of specialized data operands for the existing data types or, for the case of the floating point, the inclusion of new data types and operations for them as implemented by the TS68881 and TS68882 floating point co-processors.

The programmer's model for the TS68000 Family of microprocessors is based on sequential, non-concurrent instruction execution. This means each instruction is completely executed prior to the beginning of the next instruction. Hence, instructions do not operate concurrently in the programmer's model. Most microprocessors implement the sequential model which greatly simplifies the programmer responsibilities since sequencing control is automatic and discrete.

The TS68000 co-processor interface is designed to extend the programmer's model and it provides full support for the sequential, non-concurrent instruction execution model. Hence, instruction execution by the co-processor is assumed to not overlap with instruction execution with the main microprocessor. Yet, the TS68000 co-processor interface does allow concurrent operation when concurrency can be properly accommodated. For example, the TS68881 or TS68882 floating-point co-processor will allow the TS68020 to proceed executing instruction while the co-processor continues a floating-point operation, up to the point that the TS68020 sends another request to the co-processor. Adhering to the sequential execution model, the request to the co-processor continues a floating-point operation, up to the co-processor completes each TS68881 and TS68882 instruction before it starts the next, and the TS68020 is allowed to proceed as it can in a concurrent fashion.

co-processors are divided into two types by their bus utilization characteristics. A co-processor is a DMA co-processor if it can control the bus independent of the main processor. A co-processor is a non-DMA co-processor if it does not have the capability of controlling the bus. Both co-processor types utilize the same protocol and main processor resources. Implementation of a co-processor as a DMA or non-DMA type is based primarily on bus bandwidth of the co-processor, performance, and cost issues.

The communication protocol between the main processor and the co-processor necessary to execute a co-processor instruction is based on a group of co-processor interface registers (Table 10) which are defined for the TS68000 Family co-processor interface. The TS68020 hardware uses standard TS68000 asynchronous bus cycles to access the registers. Thus, the co-processor doesn't require a special bus hardware; the bus interface implemented by a co-processor for its interface register set must only satisfy the TS68020 address, data, and control signal timing to guarantee proper communication with the main processor. The TS68020 implements the communication protocol with all co-processors in hardware (and microcode) and handles all operations automatically so the programmer is only concerned with the instructions and data types provided by the co-processor as extensions to the TS68020 instruction set and data types.

Other microprocessors in the TS68000 Family can operate any TS68000 co-processor even though they may not have the hardware implementation of the co-processor interface as does the TS68020. Since the co-processor is operated through the co-processor interface registers which are accessed via normal asynchronous bus cycles, the co-processor may be used as a peripheral device. Software easily emulates the communication protocol by addressing the co-processor interface registers appropriately and passing the necessary commands and operands required by the co-processor.

The co-processor interface registers are implemented by the co-processor in addition to those registers implemented as extensions to the TS68020 programmer's model. For example, the TS68881 implements the co-processor interface registers shown in Table 10 and the registers in the programming model, including eight 80-bit floating-point data registers and three 32-bit control/status registers used by the TS68881 programmer.

**Table 10.** Co-processor Interface Registers

| Register | Function | R/W |
|---|---|---|
| Response | Requests Action from CPU | R |
| Control | CPU | W |
| Save | Initiate Save of Internal State | R |
| Restore | Initiate Restore of Internal State | R/W |
| Operation Word | Current Co-processor Instruction | W |
| Command Word | Co-processor Specific Command | W |
| Condition Word | Condition to be Evaluated | W |
| Operand | 32-bit Operand | R/W |
| Register Select | Specifies CPU Register or Mask | R |
| Instruction Address | Pointer to Co-processor Instruction | R/W |
| Operand Address | Pointer to Co-processor Operand | R/W |

**Table 11.** Co-processor Primitives

| |
|---|
| Processor Synchronization<br>Busy with Current Instruction<br>Proceed with Next Instruction, If No Trace<br>Service Interrupts and Re-query, If Trace Enable<br>Proceed with Execution, Condition True/False |
| Instruction Manipulation<br>Transfer Operation Word<br>Transfer Words from Instruction Stream |
| Exception Handling<br>Take Privilege Violation if S Bit Not Set<br>Take Pre-Instruction Exception<br>Take Mid-Instruction Exception<br>Take Post-Instruction Exception |

When the main processor encounters the next co-processor instruction, the main processor queries the co-processor until the co-processor is ready; meanwhile, the main processor can go on to service interrupts and do a context switch to execute other tasks, for example.

Each co-processor instruction type has specific requirements based on this simplified protocol. The co-processor interface may use as many extension words as requires to implement a co-processor instruction.

### Primitives/Response

The response register is the means by which the co-processor communicates service requests to the main processor. The content of the co-processor response register is a primitive instruction to the main processor which is read during co-processor communication by the main processor. The main processor “executes” this primitive, thereby providing the services requires by the co-processor. Table 11 summarizes the co-processor primitives that the TS68020 accepts.

## Exceptions

### Kinds of Exceptions

Exception can be generated by either internal or external causes. The externally generated exceptions are the interrupts, the bus error, and reset requests. The interrupts are requests from peripheral devices for processor action while the bus error and reset pins are used for access control and processor restart. The internally generated exceptions come from instructions, address errors, tracing, or breakpoints. The TRAP, TRAPcc, TRAPV, cpTRAPcc, CHK, CHK2, and DIV instructions can all generate exceptions as part of their execution. Tracing behaves like a very high priority, internally generated interrupt whenever it is processed. The other internally generated exceptions are caused by illegal instructions, instruction fetches from odd addresses, and privilege violations.

### Exception Processing Sequence

Exception processing occurs in four steps. During the first step, an internal copy is made of the status register. After the copy is made, the special processor state bits in the status register are changed. The S bit is set, putting the processor into supervisor privilege state. Also, the T1 and T0 bits are negated, allowing the exception handler to execute unhindered by tracing. For the reset and interrupt exceptions, the interrupt priority mask is also updated.

In the second step, the vector number of the exception is determined. For interrupts, the vector number is obtained by a processor read that is classified as an interrupt acknowledge cycle. For co-processor detected exceptions, the victor number is included in the co-processor exception primitive response. For all other exceptions, internal logic provides the vector number. This vector number is then used to generate the address of the exception vector.

The third step is to save the current processor status. The exception stack frame is created and filled on the supervisor stack. In order to minimize the amount of machine state that is saved, various stack frame sizes are used to contain the processor state depending on the type of exception and where it occurred during instruction execution. If the exception is an interrupt and the M bit is on, the M bit is forced off, and a short four word exception stack frame is saved on the master stack which indicates that the exception is saved on the interrupt stack. If the exception is a reset, the M bit is simply forced off, and the reset vector is accessed.

The TS68020 provides an extension to the exception stacking process. If the M bit in the status register is set, the master stack pointer (MSP) is used for all task related exceptions. When a non-task exception occurs (i.e., an interrupt), the M bit is cleared and the interrupt stack pointer (ISP) is used. This feature allows all the task's stack area to be carried within a single processor control block and new tasks may be initiated by simply reloading the master stack pointer and setting the M bit.

The fourth and last step of the exception processing is the same for all exceptions. The exception vector offset is determined by multiplying the vector number by four. This offset is then added to the contents of the vector base register (VBR) to determine the memory address of the exception vector. The new program counter value is fetched from the exception vector. The instruction at the address given in the exception vector is fetched, and the normal instruction decoding and execution is started.

## On-chip Instruction Cache

Studies have shown that typical programs spend most of their execution time in a few main routines or tight loops. This phenomenon is known as locality of reference, and has an impact on performance of the program. The TS68020 takes limited advantage of this phenomenon in the form of its loop mode operation which allows certain instructions, when coupled with the DBcc instruction, to execute without the overhead of instruction fetches. In effect, this is a three word cache. Although the cache hardware has been supplied in a full range of computer systems for many years, technology now allows this feature to be integrated into the microprocessor.

### TS68020 Cache Goals

There were two primary goals for the TS68020 microprocessor cache. The first design goal was to reduce the processor external bus activity. In a given TS68000 system, the TS68000 processor will use approximately 80 to 90 percent (for greater) of the available bus bandwidth. This is due to its extremely efficient perfecting algorithm and the overall speed of its internal architecture design. Thus, in an TS68000 system with more than one bus master (such as a processor and DMA device) or in a multiprocessor system, performance degradation can occur due to lack of available bus bandwidth. Therefore, an important goal for an TS68020 on-chip cache was to provide a substantial increase in the total available bus bandwidth.

The second primary design goal was to increase effective CPU throughput as larger memory sizes or slower memories increased average access time. By placing a high speed cache between the processor and the rest of the memory system, the effective access time now becomes:

$$t_{ACC} = h^{**}t_{CACHE} = (1 - h)^{*}t_{ext}$$

where $t_{ACC}$ is the effective system access time, $t_{CACHE}$ is the cache access time, $t_{ext}$ is the access time of the rest of the system, and h is the hit ratio or the percentage of time that the data is found in the cache. Thus, for a given system design, an TS68020 on-chip cache provides a substantial CPU performance increase, or allows much slower and less expensive memories to be used for the same processor performance.

The throughput increase in the TS68020 is gained in two ways. First, the TS68020 cache is accessed in two clock cycles versus the three cycles (minimum) required for an external access. Any instruction fetch that is currently resident in the cache will provide a 33% improvement over the corresponding external access.

# Preparation for Delivery

## Certificate of Compliance

Atmel offers a certificate of compliance with each shipment of parts, affirming the products are in compliance with MIL-STD-883 and guaranteeing the parameters are tested at extreme temperatures for the entire temperature range.

## Handling

MOS devices must be handled with certain precautions to avoid damage due to accumulation of static charge. Input protection devices have been designed in the chip to minimize the effect of this static buildup. However, the following handling practices are recommended:

a) Device should be handled on benches with conductive and grounded surface.

b) Ground test equipment, tools and operator

c) Do not handle devices by the leads.

d) Store devices in conductive foam or carriers.

e) Avoid use of plastic, rubber, or silk in MOS areas.

f) Maintain relative humidity above 50%, if practical.

# Package Mechanical Data

**Figure 23.** 114-lead - Ceramic Pin Grid Array

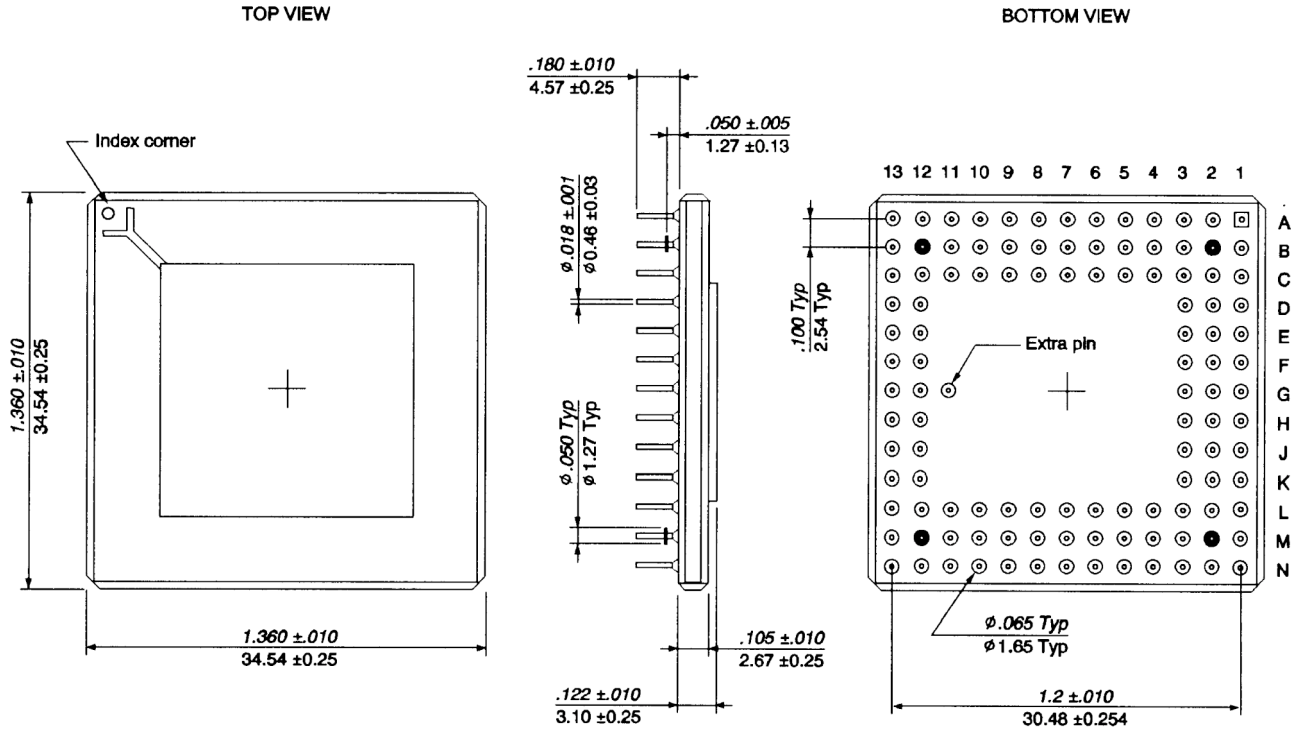
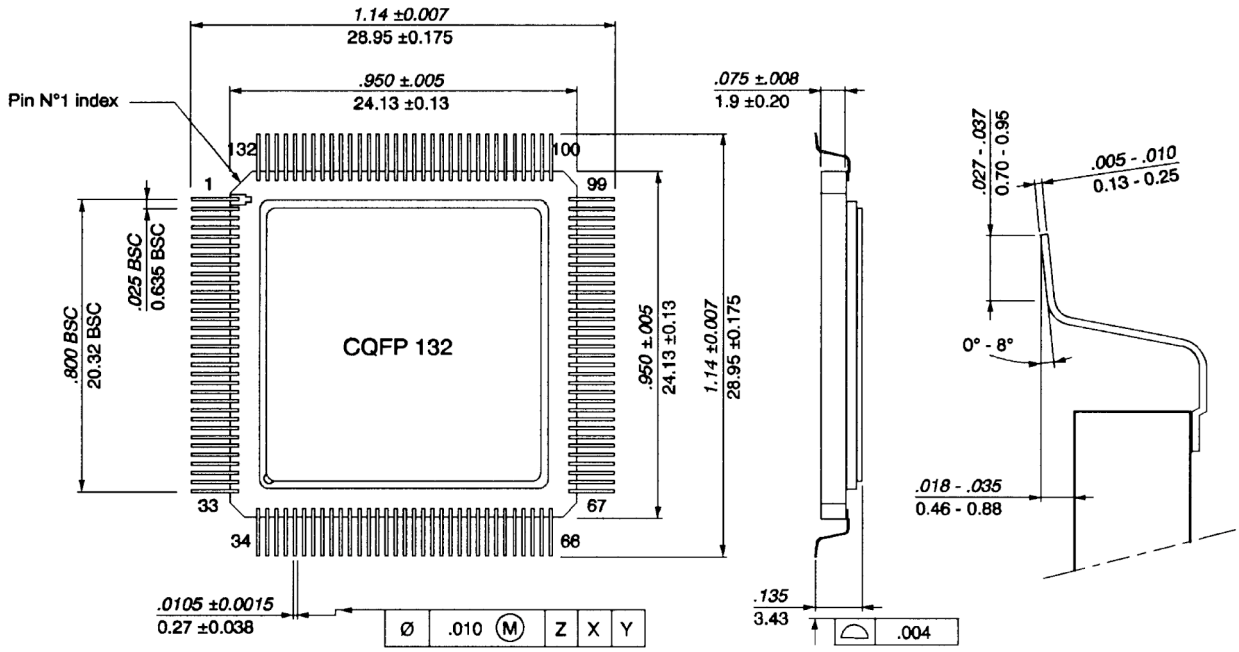



**Figure 24.** 132 Pins - Ceramic Quad Flat Pack



# Ordering Information

## Hi-REL Product

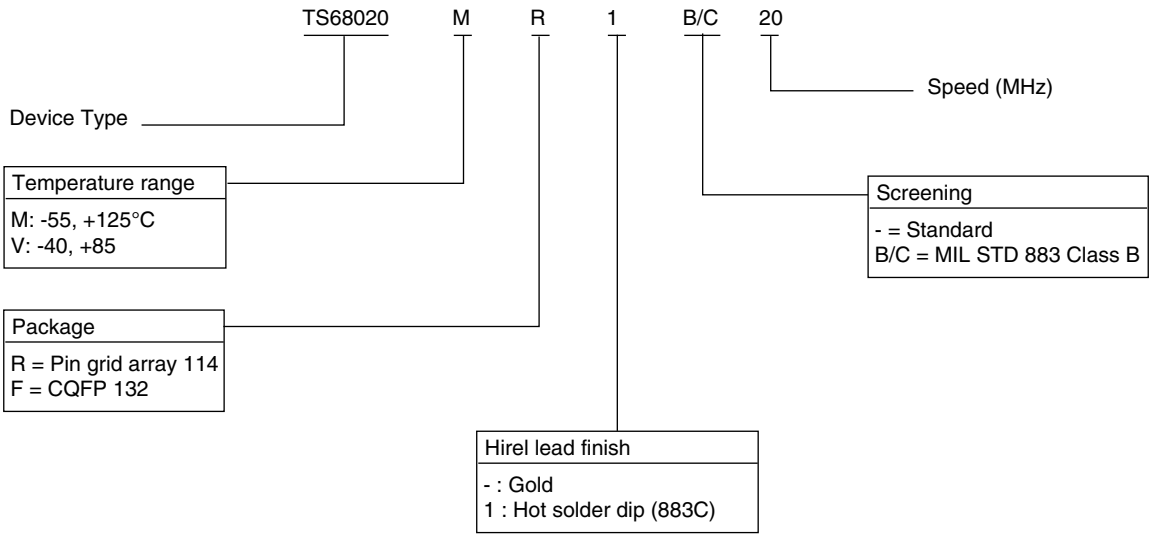
| Commercial Atmel Part-Number | Norms | Package | Temperature Range $T_c$ (°C) | Frequency (MHz) | Drawing Number |
|---|---|---|---|---|---|
| TS68020MRB/C16 | MIL-STD-883 | PGA 114 | -55/+125 | 16.67 | - |
| TS68020MR1B/C16 | MIL-STD-883 | PGA 114/tin | -55/+125 | 16.67 | - |
| TS68020MRB/C20 | MIL-STD-883 | PGA 114 | -55/+125 | 20 | - |
| TS68020MR1B/C20 | MIL-STD-883 | PGA 114/tin | -55/+125 | 20 | - |
| TS68020MRB/C25 | MIL-STD-883 | PGA 114 | -55/+125 | 25 | - |
| TS68020MR1B/C25 | MIL-STD-883 | PGA 114/tin | -55/+125 | 25 | - |
| TS68020MFB/C16 | MIL-STD-883 | CQFP 132 | -55/+125 | 16.67 | - |
| TS68020MF1B/C16 | MIL-STD-883 | CQFP 132/tin | -55/+125 | 16.67 | - |
| TS68020MFB/C20 | MIL-STD-883 | CQFP 132 | -55/+125 | 20 | - |
| TS68020MF1B/C20 | MIL-STD-883 | CQFP 132/tin | -55/+125 | 20 | - |
| TS68020MFB/C25 | MIL-STD-883 | CQFP 132 | -55/+125 | 25 | - |
| TS68020MF1B/C25 | MIL-STD-883 | CQFP 132/tin | -55/+125 | 25 | - |
| TS68020DESC02XA | DESC | PGA 114/tin | -55/+125 | 16.67 | 5962-8603202XA |
| TS68020DESC03XA | DESC | PGA 114/tin | -55/+125 | 20 | 5962-8603203XA |
| TS68020DESC04XA | DESC | PGA 114/tin | -55/+125 | 25 | 5962-8603204XA |
| TS68020DESC02XC | DESC | PGA 114 | -55/+125 | 16.67 | 5962-8603202XC |
| TS68020DESC03XC | DESC | PGA 114 | -55/+125 | 20 | 5962-8603203XC |
| TS68020DESC04XC | DESC | PGA 114 | -55/+125 | 25 | 5962-8603204XC |
| TS68020DESC02YA | DESC | CQFP 132/tin | -55/+125 | 16.67 | 5962-8603202YA |
| TS68020DESC03YA | DESC | CQFP 132/tin | -55/+125 | 20 | 5962-8603203YA |
| TS68020DESC04YA | DESC | CQFP 132/tin | -55/+125 | 25 | 5962-8603204YA |
| TS68020DESC02YC | DESC | CQFP 132 | -55/+125 | 16.67 | 5962-8603202YC |
| TS68020DESC03YC | DESC | CQFP 132 | -55/+125 | 20 | 5962-8603203YC |
| TS68020DESC04YC | DESC | CQFP 132 | -55/+125 | 25 | 5962-8603204YC |

## Standard Product

| Commercial Atmel Part-Number | Norms | Package | Temperature Range $T_c$ (°C) | Frequency (MHz) | Drawing Number |
|---|---|---|---|---|---|
| TS68020VR16 | Internal Standard | PGA 114 | -40/+85 | 16.67 | Internal |
| TS68020VR20 | Internal Standard | PGA 114 | -40/+85 | 20 | Internal |
| TS68020VR25 | Internal Standard | PGA 114 | -40/+85 | 25 | Internal |
| TS68020MR16 | Internal Standard | PGA 114 | -55/+125 | 16.67 | Internal |
| TS68020MR20 | Internal Standard | PGA 114 | -55/+125 | 20 | Internal |
| TS68020MR25 | Internal Standard | PGA 114 | -55/+125 | 25 | Internal |

## Standard Product

| Commercial Atmel Part-Number | Norms | Package | Temperature Range $T_c$ (°C) | Frequency (MHz) | Drawing Number |
|---|---|---|---|---|---|
| TS68020VF16 | Internal Standard | CQFP 132 | -40/+85 | 16.67 | Internal |
| TS68020VF120 | Internal Standard | CQFP 132 | -40/+85 | 20 | Internal |
| TS68020VF25 | Internal Standard | CQFP 132 | -40/+85 | 25 | Internal |
| TS68020MF16 | Internal Standard | CQFP 132 | -55/+125 | 16.67 | Internal |
| TS68020MF20 | Internal Standard | CQFP 132 | -55/+125 | 20 | Internal |
| TS68020MF25 | Internal Standard | CQFP 132 | -55/+125 | 25 | Internal |




Note: For availability of the different versions, contact your Atmel sales office.