TS68020

HCMOS 32-BIT VIRTUAL MEMORY MICROPROCESSOR

DESCRIPTION

The TS 68020 is the first full 32-bit implementation of the TS 68000 family of microprocessors. Using HCMOS technology, the TS 68020 is implemented with 32-bit registers and data paths, 32-bit addresses, a rich instruction set, and versatile addressing modes.

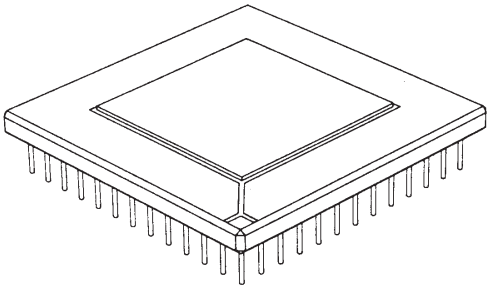
MAIN FEATURES

- Object code compatible with earlier TS 68000 microprocessors.
- Addressing mode extensions for enhanced support of high level languages.
- New bit field data type accelerates bit-oriented application, i.e. video graphics.
- Fast on-chip instruction cache speeds instructions and improves bus bandwidth.
- Coprocessor interface to companion 32-bit peripherals : TS 68881 and TS 68882 floating point coprocessors.
- Pipelined architecture with high degree of internal parallelism allowing multiple instructions to be executed concurrently.
- High performance asynchronous bus in non-multiplexed and full 32 bits.
- Dynamic bus sizing efficiently supports 8-/ 16-/ 32-bit memories and peripherals.
- Full support of virtual memory and virtual machine.
- Sixteen 32-bit general-purpose data and address registers.
- Two 32-bit supervisor stack pointers and 5 special purpose control registers.
- 18 addressing modes and 7 data types.
- 4-Gigabyte direct addressing range.
- Processor speed : 16.67 MHz - 20 MHz - 25 MHz.
- Power supply : 5.0 V_{DC} ±10 %.

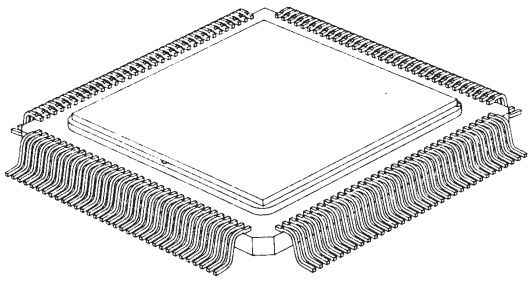
SCREENING / QUALITY

This product is manufactured in full compliance with either :

- MIL-STD-883 (class B)
- DESC 5962 - 860320
- or according to TCS standards.



R Suffix
PGA 114
Ceramic Pin Grid Array



F Suffix
CQFP 132
Ceramic Quad Flat Pack

See the ordering information page 31.

Pin connection : see page 4.

SUMMARY

A - GENERAL DESCRIPTION

INTRODUCTION

The TS 68020 is a high-performance 32-bit microprocessor. It is the first microprocessor to have evolved from a 16-bit machine to a full 32-bit machine that provides 32-bit address and data buses as well as 32-bit internal structures. Many techniques were utilized to improve performance and at the same time maintain compatibility with other processors of the TS 68000 Family. Among the improvements are new addressing modes which better support high-level language structures, an expanded instruction set which provides 32-bit operations for the limited cases not supported by the TS 68000 and sereval new instructions which support new data types. For special-purpose applications when a general-purpose processor alone is not adequate, a coprocessor interface is provided.

The TS 68020 is a high-performance microprocessor implemented in HCMOS, low power, small geometry process. This process allows CMOS and HMOS (high density NMOS) gates to be combined on the same device. CMOS structures are used where speed and low power is required, and HMOS structures are used where minimum silicon area is desired. This technology enables the TS 68020 to be very fast while consuming less power (less than 1.5 watts) and still have a reasonably small die size. It utilizes about 190.000 transistors, 103.000 of which are actually implemented. The package is a pin-grid array (PGA) with 114 pins, arranged 13 pins on a side with a depopulated center and 132 pins ceramic quad flat pack.

Figure 1 is a block diagram of the TS 68020. The processor can be divided into two main sections : the bus controller and the micromachine. This division reflects the autonomy with which the sections operate.

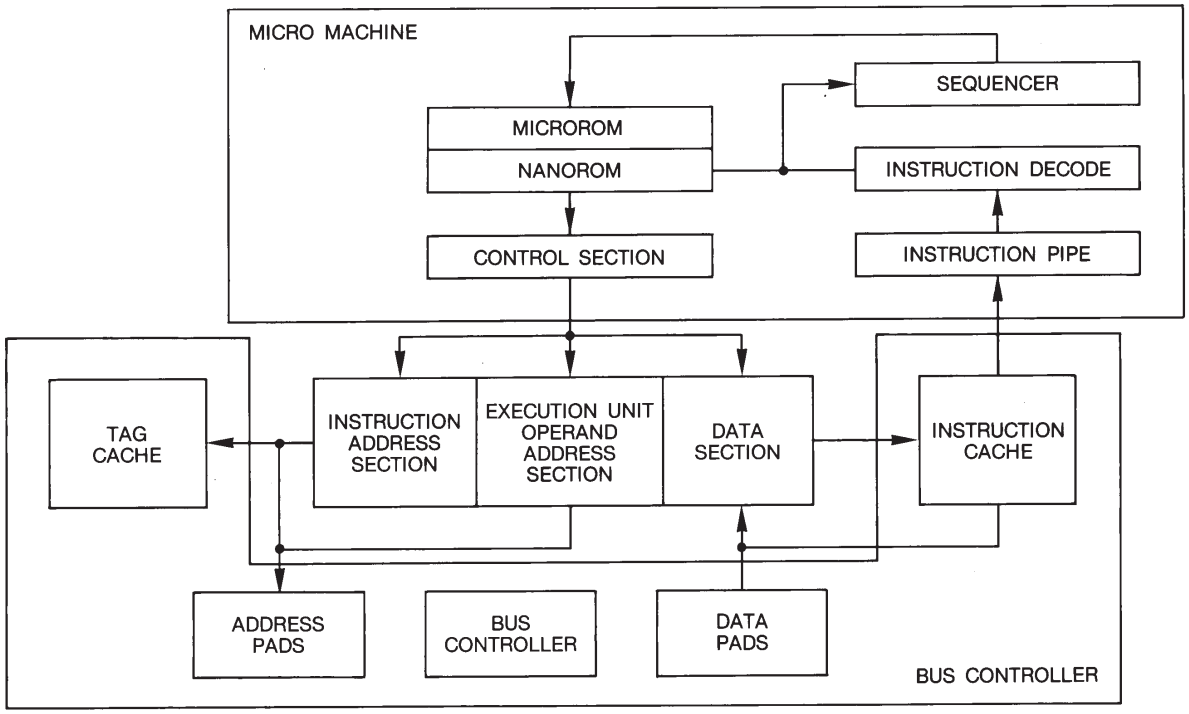


Figure 1 : TS 68020 block diagram.

The bus controller consists of the address and data pads and multiplexers required to support dynamic bus sizing, a macro bus controller which schedules the bus cycles on the basis of priority with two state machines (one to control the bus cycles for operated accesses and the other to control the bus cycles for instruction accesses), and the instruction cache with its associated control.

The micromachine consists of an execution unit, nanorom and microrom storage, an instruction decoder, an instruction pipe, and associated control sections. The execution unit consists of an address section, an operand address section, and a data section. Microcode control is provided by a modified two-level store of microrom and nanorom. Programmed logical arrays (PLAs) are used to provide instruction decode and sequencing information. The instruction pipe and other individual control sections provide the secondary decode of instructions and generated the actual control signals that result in the decoding and interpretation of nanorom and microrom information.

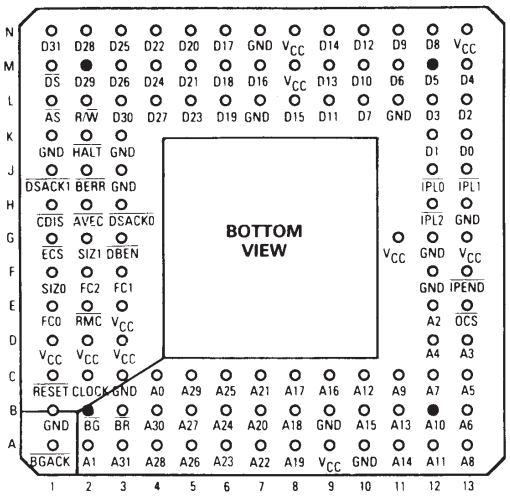


Figure 2.1 : PGA terminal designation.

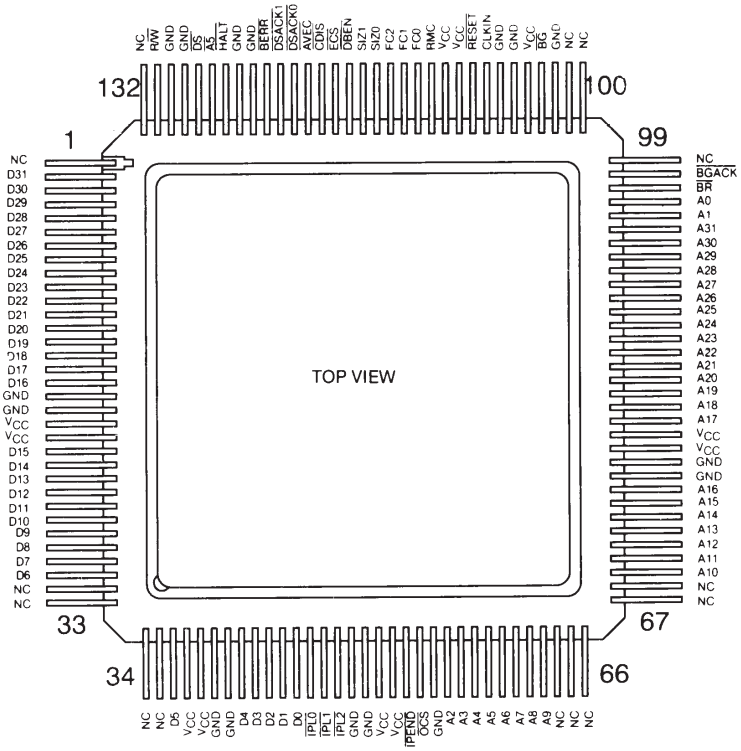


Figure 2.2 : CQFP terminal designation.

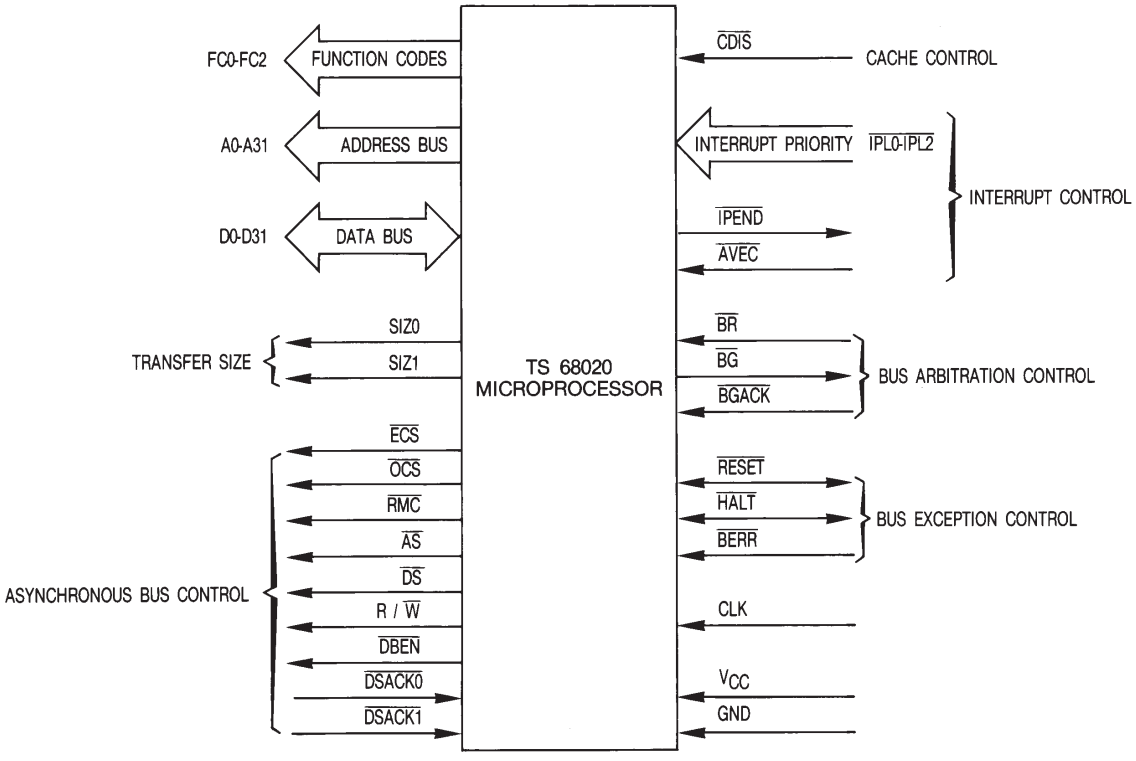


Figure 3 : Functional signal groups.

SIGNAL DESCRIPTION

Figure 3 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

Table 1 - Signal index

Signal Name	Mnemonic	Function
Address Bus	A0-A31	32-bit address bus used to address any of 4,294,967,296 bytes.
Data Bus	D0-D31	32-bit data bus used to transfer 8, 16, 24 or 32 bits of data per bus cycle.
Function Codes	FC0-FC2	3-bit function case used to identify the adress space of each bus cycle.
Size	SIZ0 / SIZ1	Indicates the number of bytes remaining to be transferred for this cycle. These signals, together with A0 and A1, define the active sections of the data bus.
Read-Modify-Write Cycle	$\overline{RMC}$	Provides an indicator that the current bus cycle is part of an indivisible read-modify-write operation.
External Cycle Start	$\overline{ECS}$	Provides an indication that a bus cycle is beginning.
Operand Cycle Start	$\overline{OCS}$	Identical operation to that of ECS except that OCS is asserted only during the first bus cycle of an operand transfer.
Address Strobe	$\overline{AS}$	Indicates that a valid address is on the bus.
Data Strobe	$\overline{DS}$	Indicates that valid data is to be placed on the data bus by an external device or has been placed on the data bus by the TS 68020.
Read / Write	R / $\overline{W}$	Defines the bus transfer as an MPU read or write.
Data Buffer Enable	$\overline{DBEN}$	Provides an enable signal for external data buffers.
Data Transfert and Size Acknowledge	$\overline{DSACK0}$ / $\overline{DSACK1}$	Bus response signals that indicate the requested data transfer operation is completed. In addition, these two lines indicate the size of the external bus port on a cycle-by-cycle basis.
Cache Disable	$\overline{CDIS}$	Dynamically disables the on-chip cache to assist emulator support.
Interrupt Priority Level	$\overline{IPL0}$-$\overline{IPL2}$	Provides an encoded interrupt level to the processor.
Autovector	$\overline{AVEC}$	Requests an autovector during an interrupt acknowledge cycle.
Interrupt Pending	$\overline{IPEND}$	Indicates that an interrupt is pending.
Bus Request	$\overline{BR}$	Indicates that an external device requires bus mastership.
Bus Grant	$\overline{BG}$	Indicates that an external device may assume bus mastership.
Bus Grant Acknowledge	$\overline{BGACK}$	Indicates that an external device has assumed bus mastership.
Reset	$\overline{RESET}$	System reset.
Halt	$\overline{HALT}$	Indicates that the processor should suspend bus activity.
Bus Error	$\overline{BERR}$	Indicates an invalid or illegal bus operation is being attempted.
Clock	$\overline{CLK}$	Clock input to the processor.
Power Supply	V_{CC}	+5 volt ±10 % power supply.
Ground	GND	Ground connection.

B - DETAILED SPECIFICATIONS

1 - SCOPE

This drawing describes the specific requirements for the microprocessor 68020, 16.67 MHz and 20 MHz, in compliance with MIL-STD-883 class B.

2 - APPLICABLE DOCUMENTS

2.1 - MIL-STD-883

1) MIL-STD-883 : test methods and procedures for electronics.
2) MIL-PRF-38535 appendix A : general specifications for microcircuits.
3) Desc Drawing 5962 - 860320xxx.

3 - REQUIREMENTS

3.1 - General

The microcircuits are in accordance with the applicable document and as specified herein.

3.2 - Design and construction

3.2.1 - Terminal connections

Depending on the package, the terminal connections shall be is shown in Figures 2.1 and 2.2.

3.2.2 - Lead material and finish

Lead material and finish shall be any option of MIL-STD-1835.

3.2.3 - Package

The macrocircuits are packaged in hermetically sealed ceramic packages which are conform to case outlines of MIL-STD-1835 (when defined) :

- 114-PIN SQ.PGA UP PAE outline,
- 132-PIN Ceramic Quad Flat Pack CQFP,
- OVCC (on request).

The precise case outlines are described on Figures 9.1 and 9.2.

3.3 - Electrical characteristics

3.3.1 - Absolute maximum ratings (see Table 2)

Table 2

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_j	Junction temperature			+160	°C
T_{leads}	Lead temperature	Max 5 sec. soldering		+270	°C

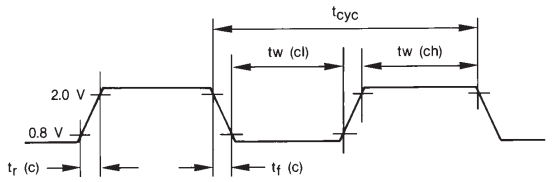
3.3.2 - Recommended condition of use

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

Table 3

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		see Note		Ω
C_L	Output loading capacitance			see Note	pF
$t_r(c) - t_f(c)$	Clock rise time (see Figure 4))	68020-16		5	ns
		68020-20		5	
		68020-25		4	
f_C	Clock frequency (see Figure 4)	68020-16	8	16.67	MHz
		68020-20	12.5	20	
		68020-25	12.5	25	
t_{cyc}	Cycle time (see Figure 4)	68020-16	60	125	ns
		68020-20	50	80	
		68020-25	40	80	
$t_W(CL)$	Clock pulse width low (see Figure 4)	68020-16	24	95	ns
		68020-20	20	54	
		68020-25	19	61	
$t_W(CH)$	Clock pulse width high (see Figure 4)	68020-16	24	95	ns
		68020-20	20	50	
		68020-25	19	61	
Note : 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 advises 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}).



Note : Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, 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 0.8 volt and 2.0 volts.

Figure 4 : Clock input timing diagram.

3.4 - Thermal characteristics (at 25°C)

Table 4

Package	Symbol	Parameter	Value	Unit
PGA 114	θ JA	Thermal resistance - Ceramic Junction to Ambient	26	°C/W
	θ JC	Thermal resistance - Ceramic Junction to Case	5	°C/W
CQFP 132	θ JA	Thermal resistance - Ceramic Junction to Ambient	34	°C/W
	θ 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 :

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

TA = Ambient Temperature, °C

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

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

$P_{INT} = I_{CC} \times V_{CC}$, 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 reliationship 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 :

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

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) iteratively for any value of T_A.

The total thermal resistance of a package (θ_{JA}) can be separated into two components, θ_{JC} and θ_{CA}, representing the barrier to heat flow from the semiconductor junction to the package (case), surface (θ_{JC}) and from the case to the outside ambient (θ_{CA}). These terms are related by the equation :

$$\theta_{JA} = \theta_{JC} + \theta_{CA} \quad (4)$$

θ_{JC} is device related and cannot be influenced by the user. However, θ_{CA} is user dependent and can be minimized by such thermal management techniques as heat sinks, ambient air cooling and thermal convection. Thus, good thermal management on the part of the user can significantly reduce θ_{CA} so that θ_{JA} approximately equals θJ_C. Substitution of θ_{JC} for θ_{JA} in equation (1) will result in a lower semiconductor junction temperature.

3.5 · Mechanical and environment

The microcircuits shall meet all mechanical environmental requirements of MIL-STD-883 for class B devices.

3.6 · Marking

The document where are defined the marking are identified in the related reference documents. Each microcircuit are legible and permanently marked with the following information as minimum :

3.6.1 · Thomson logo

3.6.2 · Manufacturer's part number

3.6.3 · Class B identification

3.6.4 · Date-code of inspection lot

3.6.5 · ESD identifier if available

3.6.6 · Country of manufacturing

4 · QUALITY CONFORMANCE INSPECTION

4.1 · DESC / MIL-STD-883

Is in accordance with MIL-M-38510 and method 5005 of MIL-STD-883. Group A and B inspections are performed on each production lot. Group C and D inspection are performed on a periodical basis.

5 · ELECTRICAL CHARACTERISTICS

5.1 · General requirements

All static and dynamic electrical characteristics specified and the relevant measurement conditions are given below. For inspection purpose, refer to relevant specification :

– DESC see § 4.1

(last issue on request to our marketing services).

Table 5 : Static electrical characteristics for all electrical variants. See § 5.2.
Table 6 : Dynamic electrical characteristics for 68020-16 (16.67 MHz), 68020-20 (20 MHz) and 68020-25 (25 MHz). See § 5.3.

For static characteristics, test methods refer to clause 5.4 hereafter of this specification (Table 7).

For dynamic characteristics (Table 6), test methods refer to IEC 748-2 method number, where existing.

Indication of «min.» or «max.» in the column «test temperature» means minimum or maximum operating temperature.

5.2 · Static characteristics

Table 5

V_{CC} = 5.0 V_{dc} ± 10 % ; GND = 0 V_{dc} ; T_C = −55 / +125°C or −40 / +85°C (See Figure 3 to Figure 7)

Symbol	Parameter	Condition	Min	Max	Units
I_{CC}	Maximum supply current	V_{CC} = 5.25 V T_{case} −55°C to +25°C		333	mA
I_{CC}	Maximum supply current	V_{CC} = 5.25 V T_{case} = 125°C		207	mA
V_{IH}	High level input voltage	V_O = 0.5 V or 2.4 V V_{CC} = 4.75 V to 5.25 V	2.0	V_{CC}	V
V_{IL}	Low level input voltage	V_O = 0.5 V or 2.4 V V_{CC} = 4.75 V to 5.25 V	−0.5	0.8	V
V_{OH}	High level output voltage All outputs	I_{OH} = 400 µA	2.4		V
V_{OL}	Low level output voltage Outputs A0-A31, FC0-FC2, D0-D31, SIZ0-SIZ1, $\overline{BG}$	I_{OL} = 3.2 mA Load circuit as Figure 7 R = 1.22 kΩ		0.5	V
V_{OL}	Low level output voltage Outputs $\overline{AS}$, $\overline{DS}$, $\overline{RMC}$, R/$\overline{W}$, $\overline{DBEN}$, $\overline{IPEND}$	I_{OL} = 5.3 mA Load circuit as Figure 7 R = 740 Ω		0.5	V
V_{OL}	Low level output voltage Outputs $\overline{ECS}$, $\overline{OCS}$	I_{OL} = 2.0 mA Load circuit as Figure 7 R = 2 kΩ		0.5	V
V_{OL}	Low level output voltage Outputs $\overline{HALT}$, $\overline{RESET}$	I_{OL} = 10.7 mA Load circuit as Figure 5 and Figure 6		0.5	V
$\|I_{IN}\|$	Input leakage current (high and low state)	−0.5 V ⩽ V_{IN} ⩽ V_{CC} (max)		2.5	µA
$\|I_{OHZ}\|$	High level leakage current at three-state outputs Outputs A0-A31, AS, $\overline{DBEN}$, $\overline{DS}$, D0-D31, R/$\overline{W}$, FC0-FC2, $\overline{RMC}$, SIZ0-SIZ1	V_{OH} = 2.4 V		2.5	µA
$\|I_{OLZ}\|$	Low level leakage current at three-state outputs Outputs A0-A31, $\overline{AS}$, $\overline{DBEN}$, $\overline{DS}$, D0-D31, R/$\overline{W}$, FC0-FC2, $\overline{RMC}$, SIZ0-SIZ1	V_{OL} = 0.5 V		2.5	µA
I_{OS}	Output short-circuit current (Any output)	V_{CC} = 5.25 V V_O = 0 V (Pulsed. Duration 1 ms Duty Cycle 10:1)		200	mA

5.3 · 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 4.75 V to 5.25 V V_{IL} = 0.5 V and V_{IH} = 2.4 V (See also note 12 and 13).

The INTERVAL numbers refer to the timing diagrams. See Figures 4, 8, and 9.

Table 6

Symbol	Parameter	Interval N°	68020-16 Min	68020-16 Max	68020-20 Min	68020-20 Max	68020-25 Min	68020-25 Max	Unit	Notes
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 invalid	25	15		10		5		ns	6
t_{DNDBN}	$\overline{DS}$ negated to $\overline{DBEN}$ negated (write)	25A	15		10		5		ns	9
t_{DVSA}	Data out valid to $\overline{DS}$ asserted (write)	26	15		10		5		ns	6
t_{DICL}	Data in valid to clock low (data setup)	27	5		5		5		ns	
t_{BELCL}	Late $\overline{BERR}$/$\overline{HALT}$ asserted to clock low setup time	27A	20		15		10		ns	
t_{SNDN}	$\overline{AS}$, $\overline{DS}$ negated to $\overline{DSACK}$x/$\overline{BERR}$/ $\overline{HALT}$/$\overline{AVEC}$ negated	28	0	80	0	65	0	50	ns	

Table 6 (continued)

Symbol	Parameter	Interval N°	68020-16 Min	68020-16 Max	68020-20 Min	68020-20 Max	68020-25 Min	68020-25 Max	Unit	Notes
t_{SNDI}	$\overline{DS}$ negated to data in invalid (data in hold time)	29	0		0		0		ns	6
t_{SNDIZ}	$\overline{DS}$ negated to data in high impedance	29A		60		50		40	ns	
t_{DADI}	$\overline{DSACKx}$ asserted to data in valid	31		50		43		32	ns	2, 11
t_{DADV}	$\overline{DSACKx}$ asserted to $\overline{DSACKx}$ valid ($\overline{DSACK}$ asserted skew)	31A		15		10		10	ns	3, 11
t_{HRrf}	$\overline{RESET}$ input transition time	32		1.5		1.5		1.5	Clks	
t_{CLBA}	Clock low to $\overline{BG}$ asserted	33	0	30	0	25	0	20	ns	
t_{CLBN}	Clock low to $\overline{BG}$ negated	34	0	30	0	25	0	20	ns	
t_{BRAGA}	$\overline{BR}$ asserted to $\overline{BG}$ asserted (RMC not asserted)	35	1.5	3.5	1.5	3.5	1.5	3.5	Clks	11
t_{GAGN}	$\overline{BGACK}$ asserted to $\overline{BG}$ negated	37	1.5	3.5	1.5	3.5	1.5	3.5	Clks	11
t_{GABRN}	$\overline{BGACK}$ asserted to $\overline{BR}$ negated	37A	0	1.5	0	1.5	0	1.5	Clks	11
t_{GN}	$\overline{BG}$ width negated	39	90		75		60		ns	11
t_{GA}	$\overline{BG}$ width asserted	39A	90		75		60		ns	
t_{CHDAR}	Clock high to $\overline{DBEN}$ asserted (read)	40	0	30	0	25	0	20	ns	
t_{CLDNR}	Clock low to $\overline{DBEN}$ negated (read)	41	0	30	0	25	0	20	ns	
t_{CLDAW}	Clock low to $\overline{DBEN}$ asserted (write)	42	0	30	0	25	0	20	ns	
t_{CHDNW}	Clock high to $\overline{DBEN}$ negated (write)	43	0	30	0	25	0	20	ns	
t_{RADA}	R/W low to $\overline{DBEN}$ asserted (write)	44	15		10		10		ns	6
t_{DA}	$\overline{DBEN}$ width asserted Read Write	45	60 120		50 100		40 80		ns ns	5 5
t_{RWA}	R/$\overline{W}$ width asserted (write or read)	46	150		125		100		ns	
t_{AIST}	Asynchronous input setup time	47A	5		5		5		ns	11
t_{AIHT}	Asynchronous input hold time	47B	15		15		10		ns	11
t_{DABA}	$\overline{DSACKx}$ asserted to BERR/HALT asserted	48		30		20		18	ns	4, 11
t_{DOCH}	Data out hold from clock high	53	0		0		0		ns	
t_{BNHN}	$\overline{BERR}$ negated to $\overline{HALT}$ negated (rerun)		0		0		0		ns	
f	Frequency of operation		8.0	16.67	12.5	20.0	12.5	25	MHz	
t_{RADC}	R/W asserted to data bus impedance change	55	30		25		20			11
t_{HRPW}	$\overline{RESET}$ pulse width (reset instruction)	56	512		512		512		Clks	11
t_{BNHN}	$\overline{BERR}$ negated to $\overline{HALT}$ negated (rerun)	57	0		0		0		ns	11
t_{GANBD}	$\overline{BGACK}$ negated to bus driven	58	1		1		1		Clks	10, 11
t_{GNBD}	$\overline{BG}$ negated to bus driven	59	1		1		1		Clks	10, 11

NOTES (= = INTERVAL Number)

1 - This number can be reduced to 5 nanoseconds if the strobes have equal loads.

2 - If the asynchronous setup time (= 47) requirements are satisfied, the $\overline{DSACKx}$ low to data setup time (= 31) and $\overline{DSACKx}$ low to $\overline{BERR}$ low setup time (= 48) can be ignored. The data must only satisfy the data in to clock low setup time (= 27) for the following clock cycle, $\overline{BERR}$ must only satisfy the late $\overline{BERR}$ low to clock setup time (= 27) for the following clock cycle.

3 - This parameter specifies the maximum allowable skew between $\overline{DSACK0}$ to $\overline{DSACK1}$ asserted or $\overline{DSACK1}$ to $\overline{DSACK0}$ asserted pattern = 47 must be met by $\overline{DSACK0}$ and $\overline{DSACK1}$.

4 - In the absence of $\overline{DSACKx}$, $\overline{BERR}$ is an asynchronous input using the asynchronous input setup time (= 47).

5 - $\overline{DBEN}$ may stay asserted on consecutive write cycles.

6 - Actual value depends on the clock input waveform.

7 - This pattern indicates the minimum high time for $\overline{ECS}$ and $\overline{OCS}$ in the event of an internal cache hit followed immediately by a cache miss or operand cycle.

8 - This specification guarantees operations with the 68881 coprocessor, and defines a minimum time for DS negated to AS asserted (= 13A). Without this parameter, incorrect interpretation of = 9A and = 15 would indicate that the 68020 does not meet 68881 requirements.

9 - This pattern allows the systems designer to guarantee data hold times on the output side of data buffers that have output enable signals generated with $\overline{DBEN}$.

10 - Guarantees that an alternate bus master has stopped driving the bus when the 68020 regains control of the bus after an arbitration sequence.

11 - Cannot be tested. Provided for system design purposes only.

12 - T_{case} = −55°C and +130°C in a Power off condition under Thermal soak for 4 minutes or until thermal equilibrium. Electrical parameters are tested «instant on» 100 m sec after power is applied.

13 - All outputs unload except for load capacitance. Clock = fmax,
LOW : $\overline{HALT}$, $\overline{RESET}$
HIGH : $\overline{DSACK0}$, $\overline{DSACK1}$, $\overline{CDIS}$, $\overline{IPL0}$-$\overline{IPL2}$, $\overline{DBEN}$, $\overline{AVEC}$, $\overline{BERR}$.

5.4 - Test conditions specific to the device

5.4.1 - Loading network

The applicable loading network shall be as defined in column «Test conditions» of Table 6, refering to the loading network number as shown in Figures 5, 6 and 7 below.

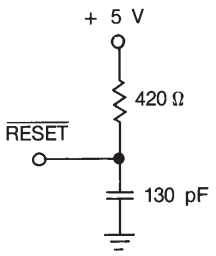


Figure 5 : RESET test load.

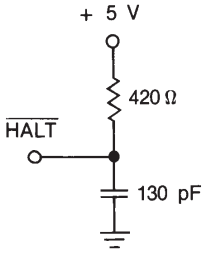


Figure 6 : HALT test load.

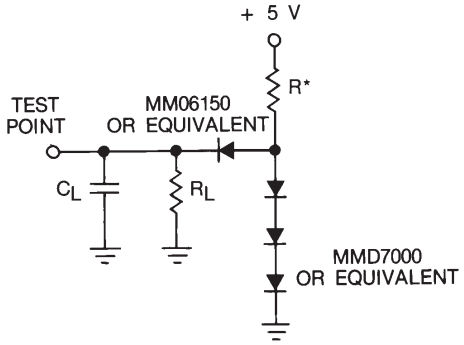


Figure 7 : Test load.

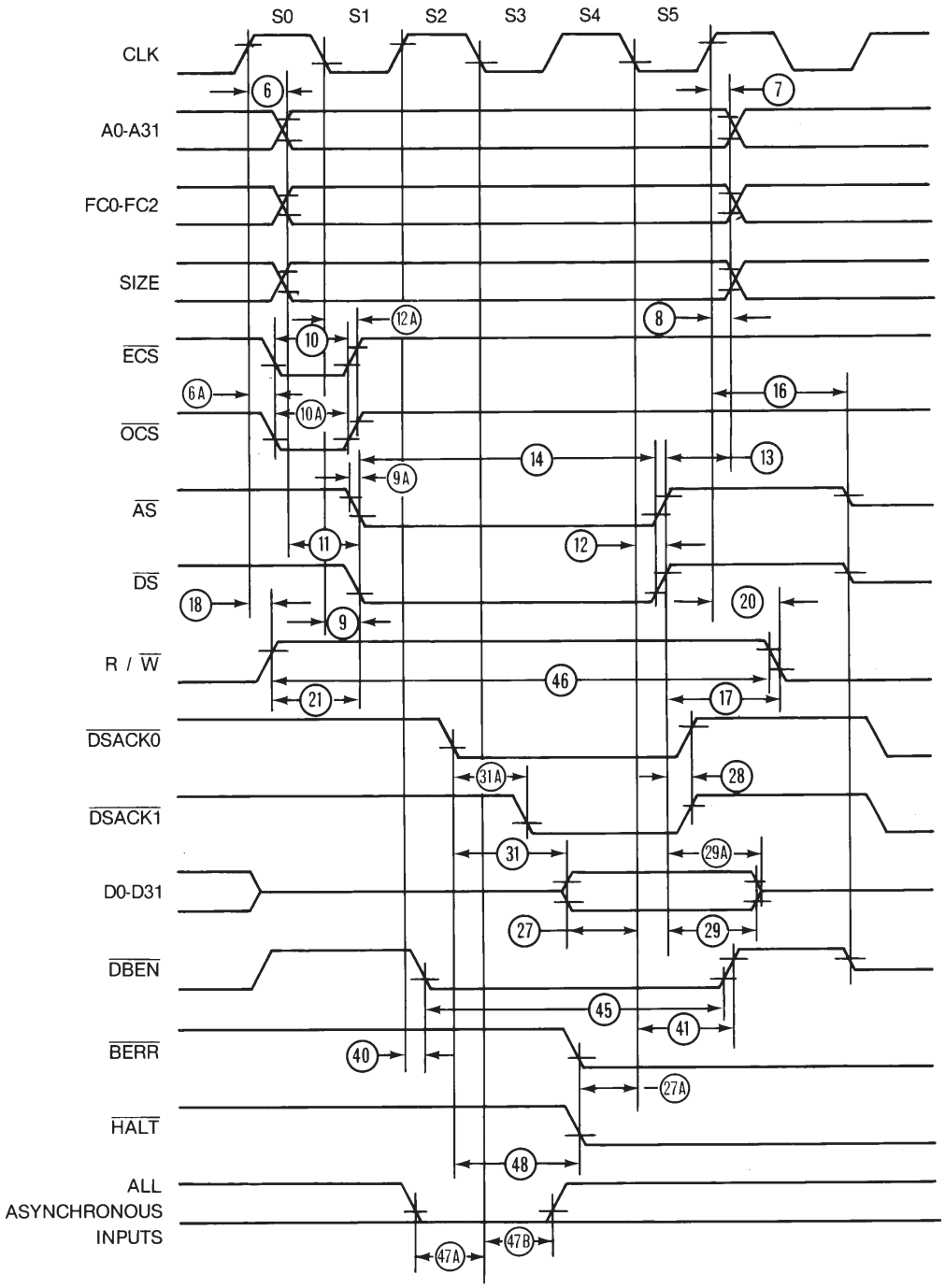
Table 7

Load NBR	Figure	R	R_L	C_L	Output Application
1	7	2 k	6.0 k	50 pF	$\overline{OCS}$, $\overline{ECS}$
2	7	1.22 k	6.0 k	130 pF	A0-A31, D0-D31, $\overline{BG}$, FC0-FC2, SIZ0-SIZ1
3	7	0.74 k	6.0 k	130 pF	$\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{RMC}$, $\overline{DBEN}$, $\overline{IPEND}$

Note : Equivalent loading may be simulated by the tester.

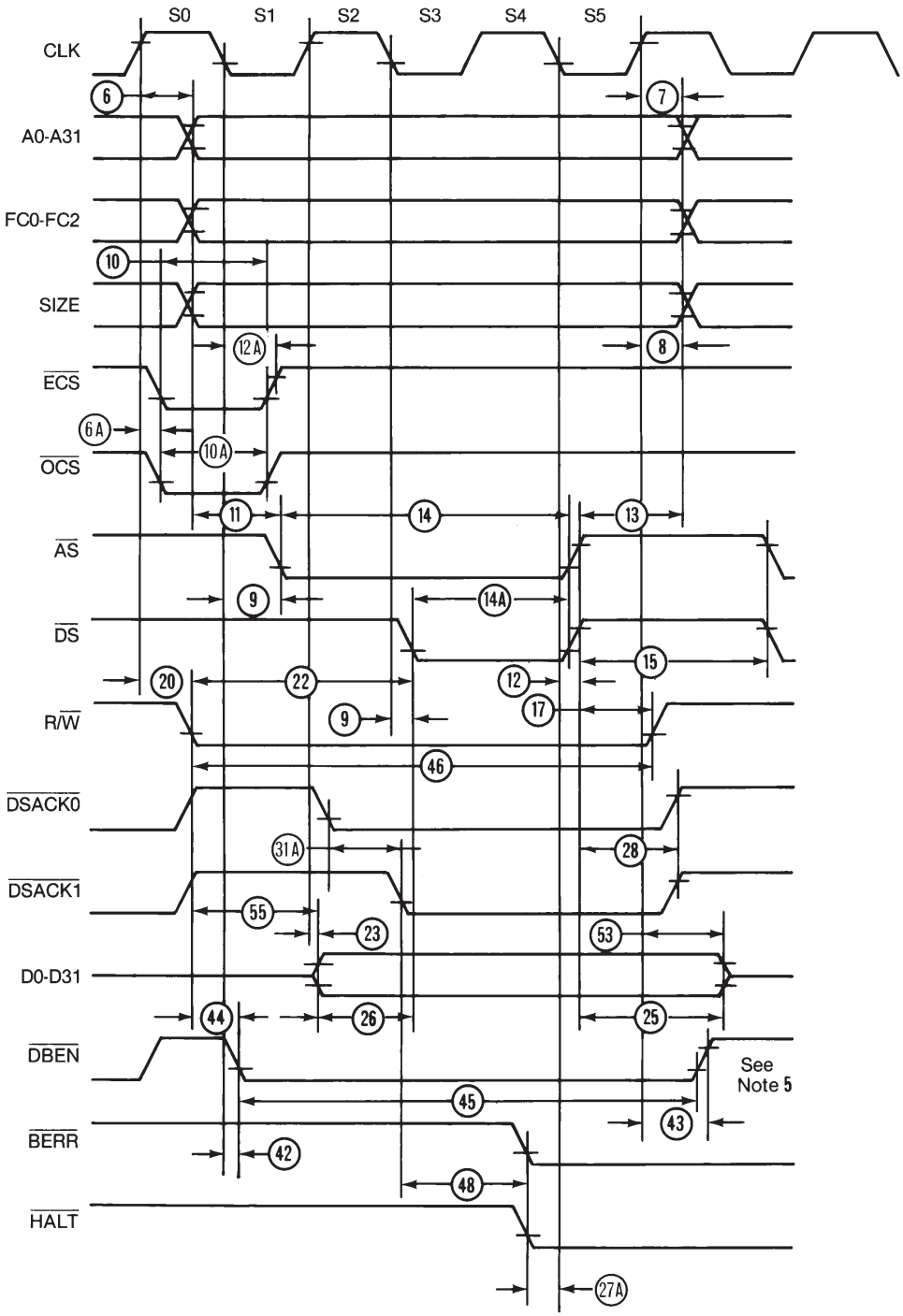
5.4.2 · Time definitions

The times specified in Table 6 as dynamic characteristics are defined in Figure 8 below, by a reference number given the column «interval N°» of the tables together with the relevant figure number.



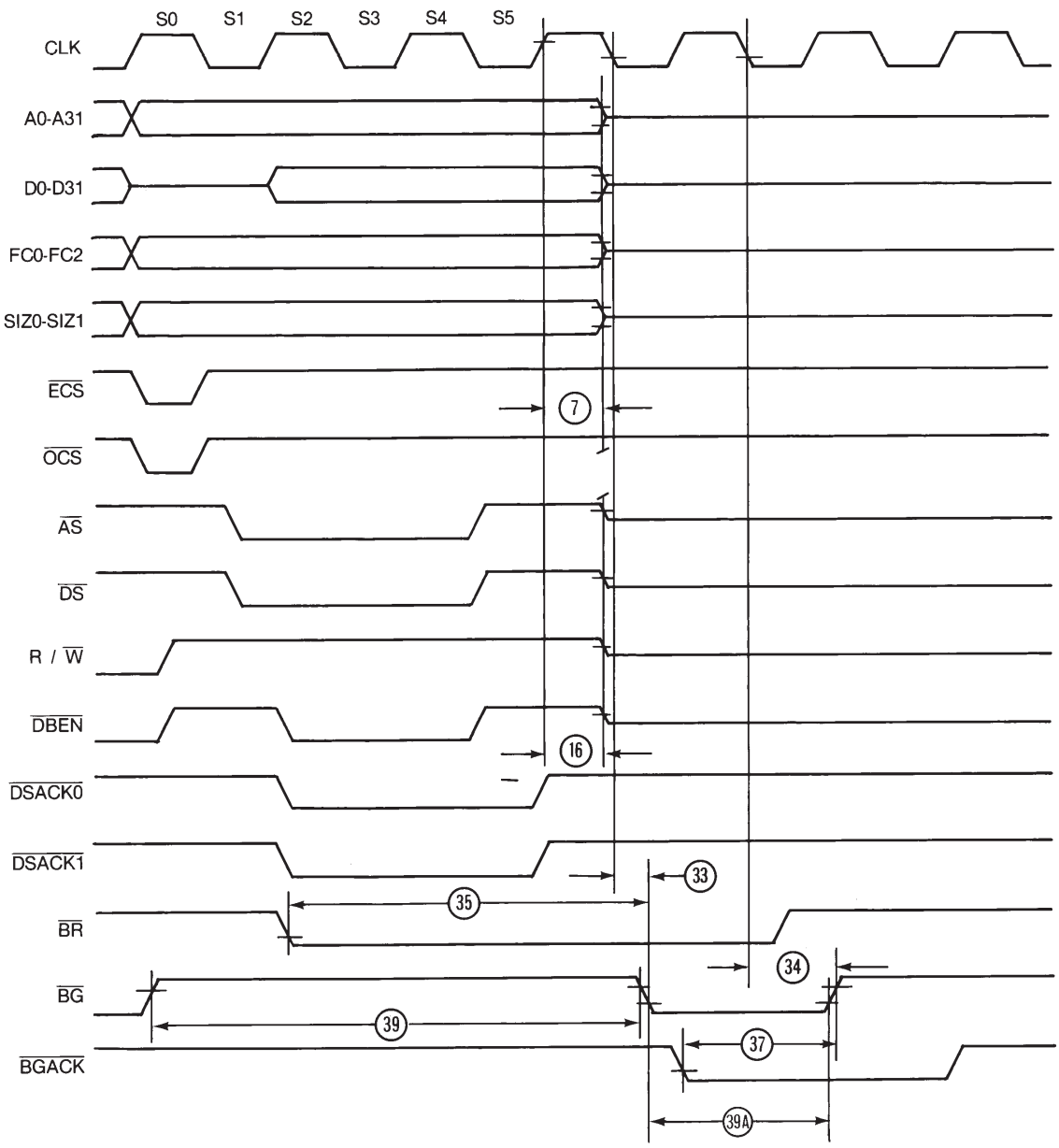
Note : Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8 volt and 2.0 volts.

Figure 8 : Read cycle timing diagram.



Note : Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, 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 0.8 volt and 2.0 volts.

Figure 8 : Write cycle timing diagram (continued).



Note : Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, 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 0.8 volt and 2.0 volts.

Figure 8 : Bus arbitration timing diagram.

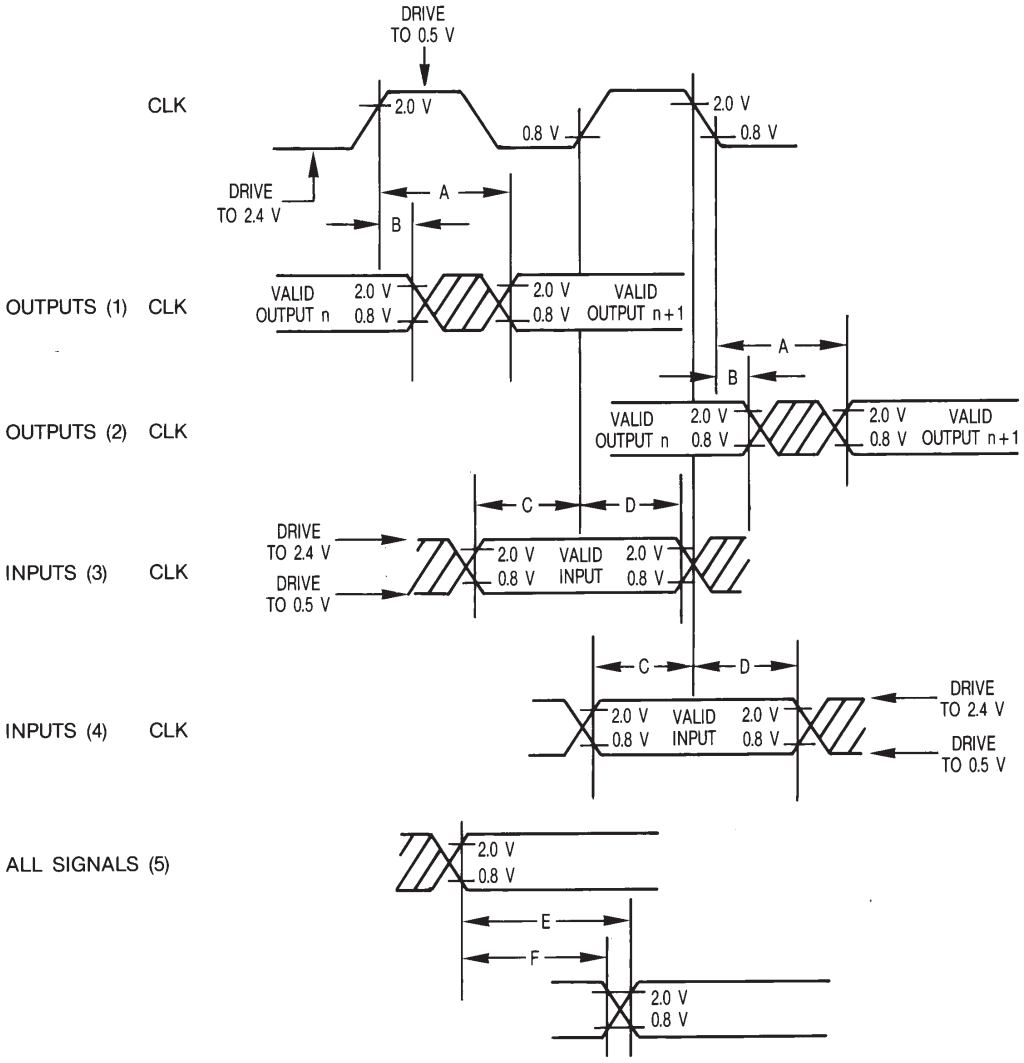
5.4.3 · Input and output signals for dynamic measurements

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 TS 68020 clock input and, possibly, relative to one or more other signals.

The measurement of the AC specifications is defined by the waveforms in Figure 9. In order to test the parameters guaranteed by TCS, inputs must be driven to the voltage levels specified in Figure 9. Outputs of the TS 68020 are specified with minimum and/or maximum limits, as appropriate, and are measured as shown. Inputs to the TS 68020 are specified with minimum and, as appropriate, maximum setup and hold times, and are measurement as shown. Finally, the measurements for signal-to-signal specifications are also shown.

Note that the testing levels used to verify conformance of the TS 68020 to the AC specifications does not affect the guaranteed DC operation of the device as specified in the DC electrical characteristics.



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 output timing is applicable to all parameters specified relative to the failing edge of the clock.
3 - This input timing is applicable to all parameters specified relative to the failing edge of the clock.
4 - This input timing is applicable to all parameters specified relative to the failing edge of the clock.
5 - This timing is applicable to all parameters specified relative to the assertion / negation of another signal.

Figure 9 : Drive levels and test points for AC specifications.

5.5 · Additional information

Additional information shall not be for any inspection purposes.

*5.5.1 · **Power considerations*** (see § 3.4)

*5.5.2 · **Capacitance (Not for inspection purposes)***

Symbol	Parameter	Test conditions	Max	Unit
C_{in}	Input capacitance	V_{in} = 0 V T_{amb} = 25°C f = 1 MHz	20	pF

Capacitance derating curves

Figures 10 to 15 inclusive show the typical derating conditions which apply. The capacitance includes any stray capacitance. The graphs may not be linear outside the range shown.

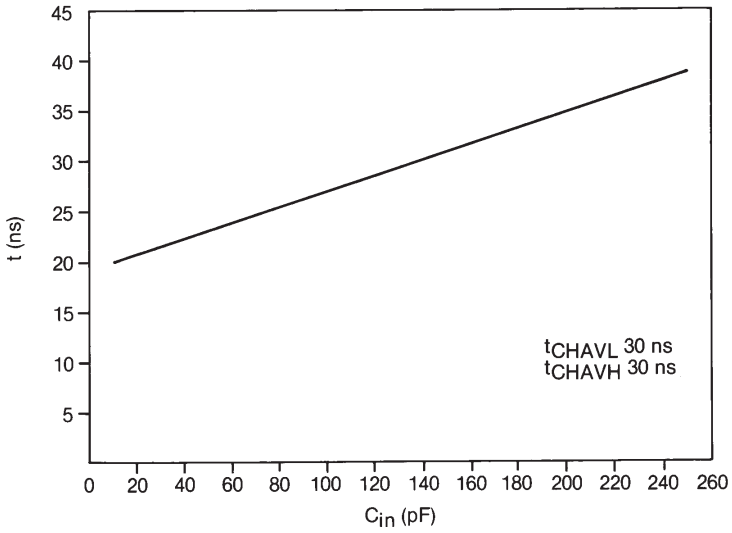


Figure 10 : Address capacitance derating curve.

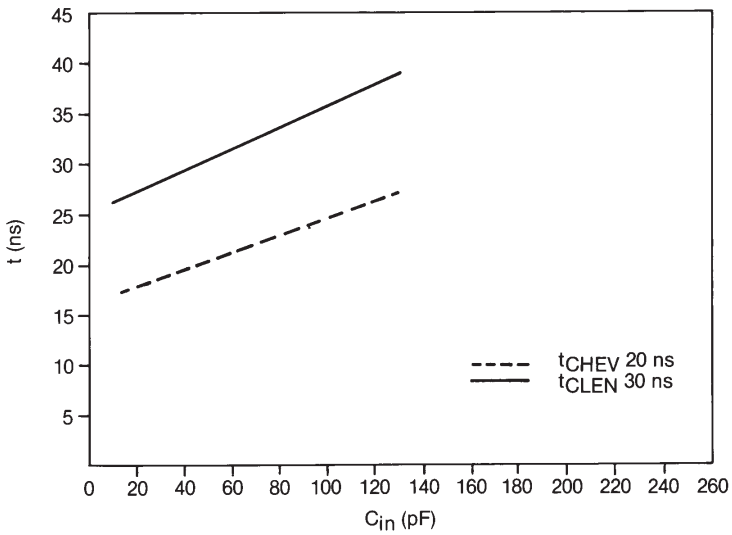


Figure 11 : ECS and OCS capacitance derating curve.

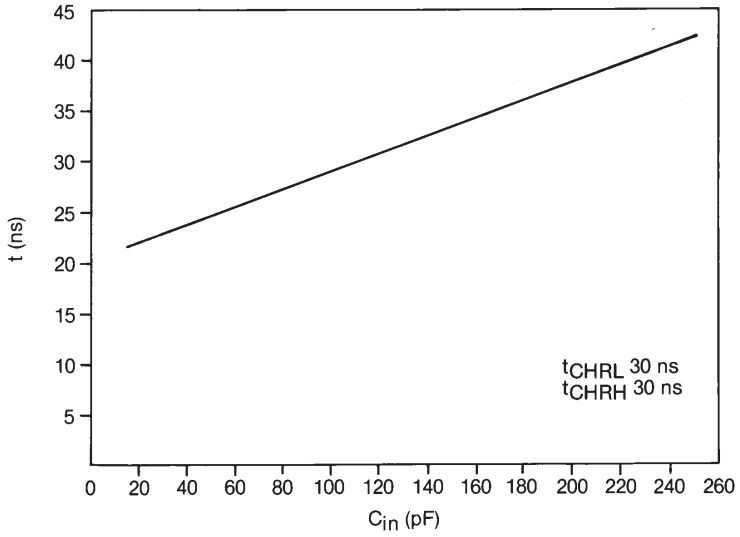


Figure 12 : R/W, FC, SIZ0-SIZ1, and RMC capacitance derating curve.

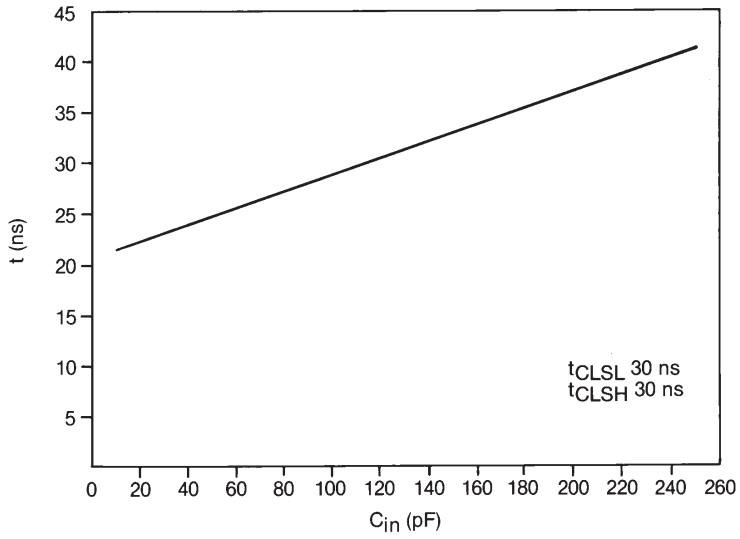


Figure 13 : DS, AS, IPEND, and BG capacitance derating curve.

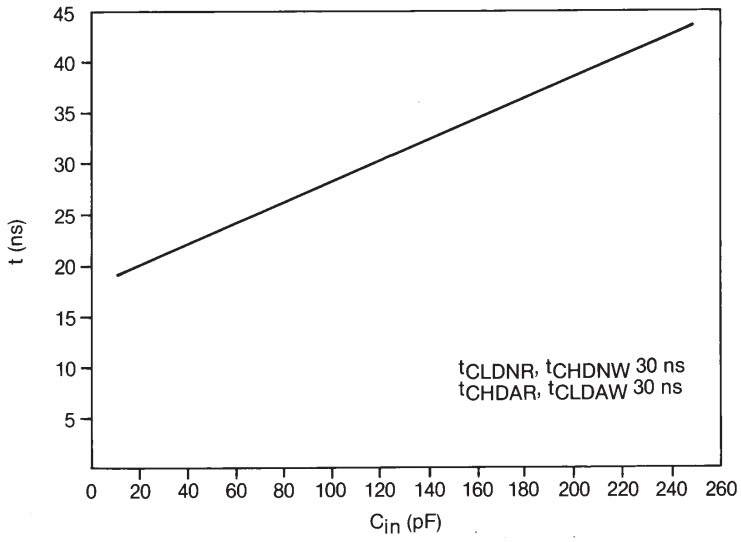


Figure 14 : DBEN capacitance derating curve.

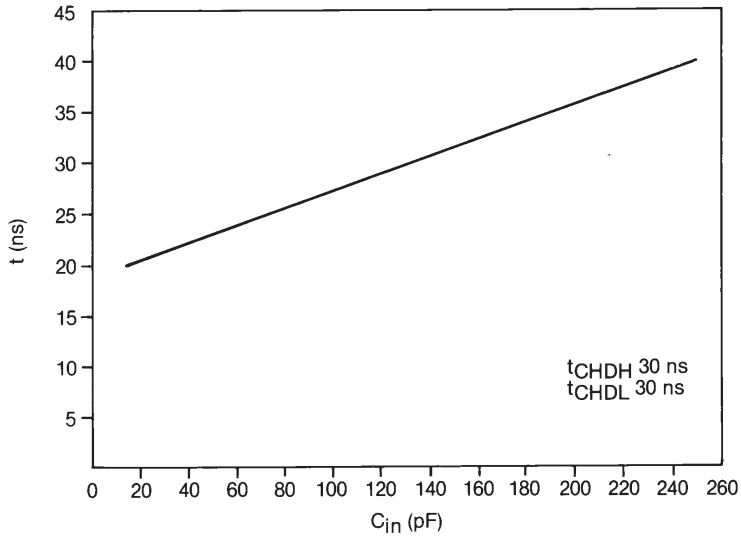


Figure 15 : Data capacitance derating curve.

6 - FUNCTIONNAL DESCRIPTION

Description of registers

As shown in the programming models (Figures 16 and 17) the TS 68020 has sixteen 32-bit general-purpose registers, a 32-bit program counter, two 32-bit supervisor stack pointers, a 16-bit status register, a 32-bit vector base register, two 3-bit alternate function code registers, and two 32-bit cache handling (address and control) registers. Registers D0-D7 are used as data registers for bit and bit field (1 to 32 bit), byte (8 bit), long word (32 bit), and quad word (64 bit) operations. Registers A0-A6 and the user, interrupt, and master stack pointers are address registers that may be used as software stack pointers or base address registers. In addition, the address registers may be used for word and long word operations. All of the 16 (D0-D7, A0-A7) registers may be used as index registers.

The status register (Figure 18) contains the interrupt priority mask (three bits) as well as the condition codes : extend (X), negated (N), zero (Z), overflow (V), and carry (C). Additional control bits indicate that the processor is in the trace mode (T1 or T0), supervisor / user state (S), and master / interrupt state (M).

All microprocessors of the TS 68000 Family support instruction tracing (via the T0 status bit in the TS 68020) where each instruction executed is followed by a trap to a user-defined trace routine. The TS 68020 adds the capability to trace only the change of flow instructions (branch, jump, subroutine call and return, etc.) using the T1 status bit. These features are important for software program development and debug.

The vector base register is used to determine the runtime location of the exception vector table in memory, hence it supports multiple vector tables so each process or task can propertly manage exceptions independent of each other.

The TS 68000 Family processors distinguish address spaces as supervisor / user and program / data. These four combinations are specified by the function code pins (FC0 / FC1 / FC2) during bus cycles, indicating 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.

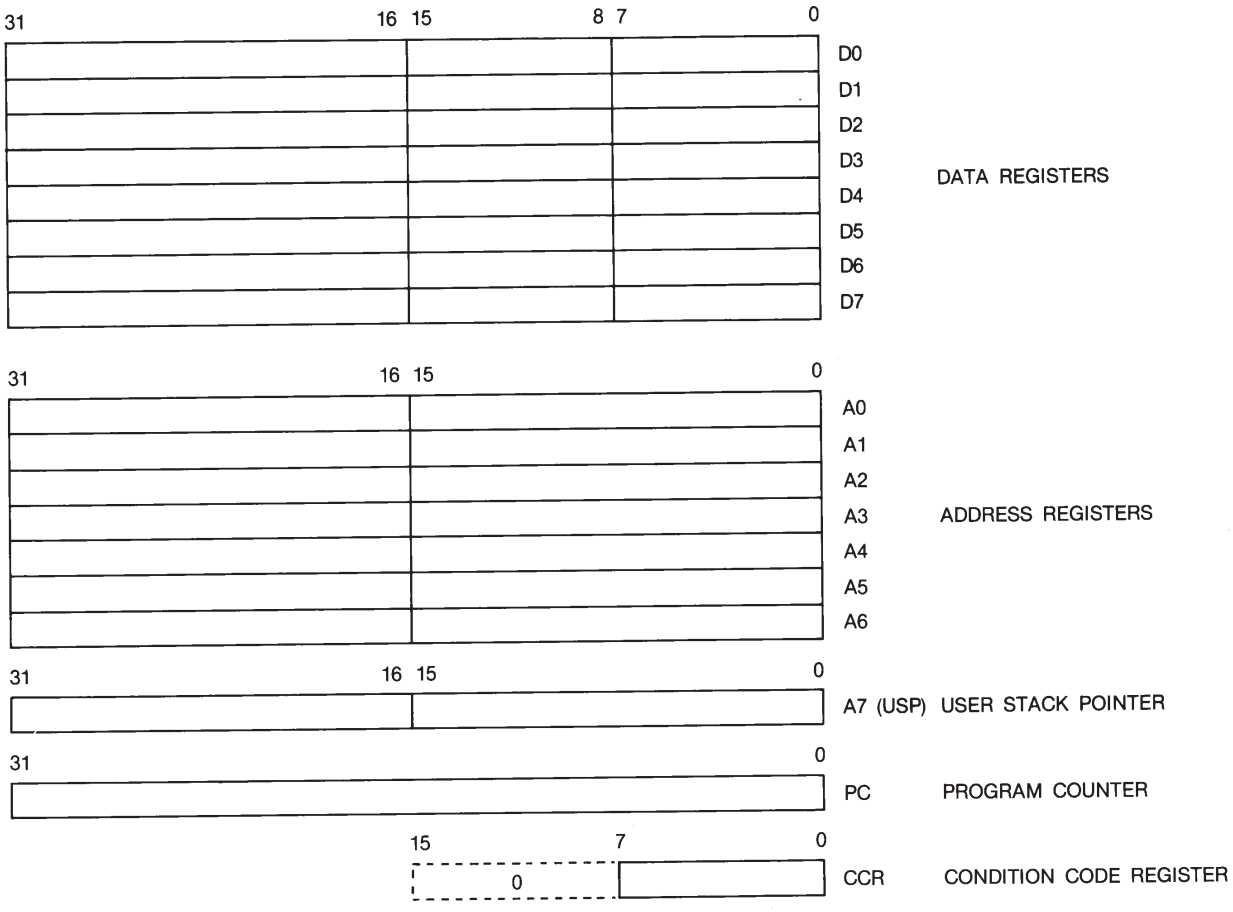


Figure 16 : User programming model.

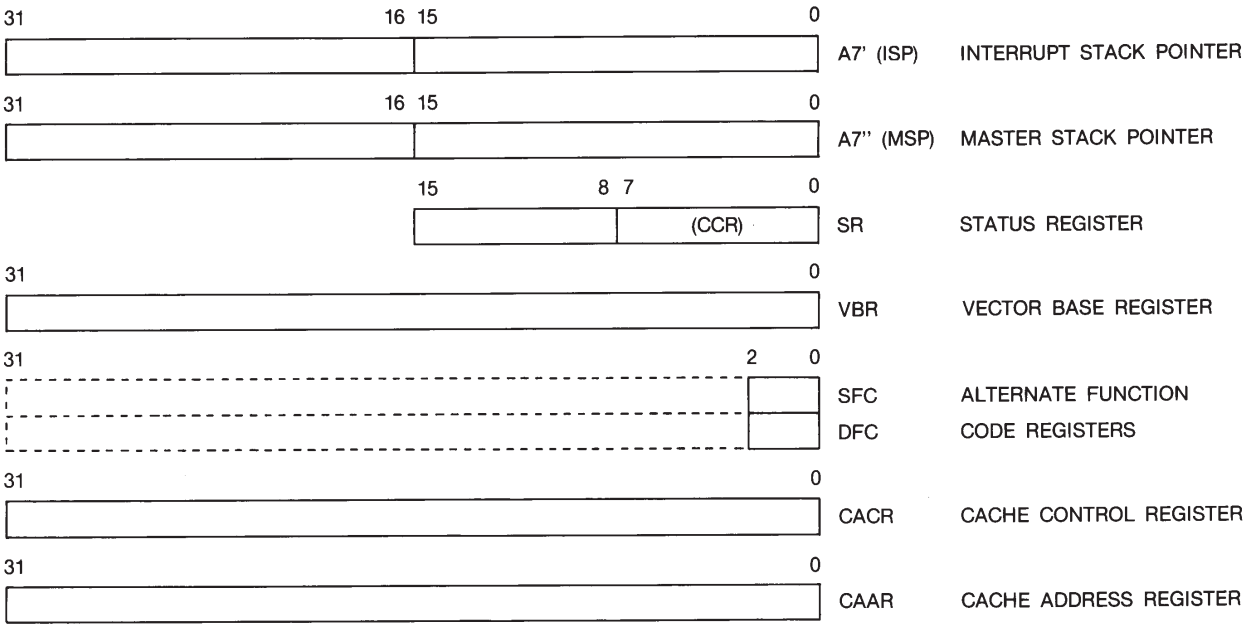


Figure 17 : Supervisor programming model supplement.

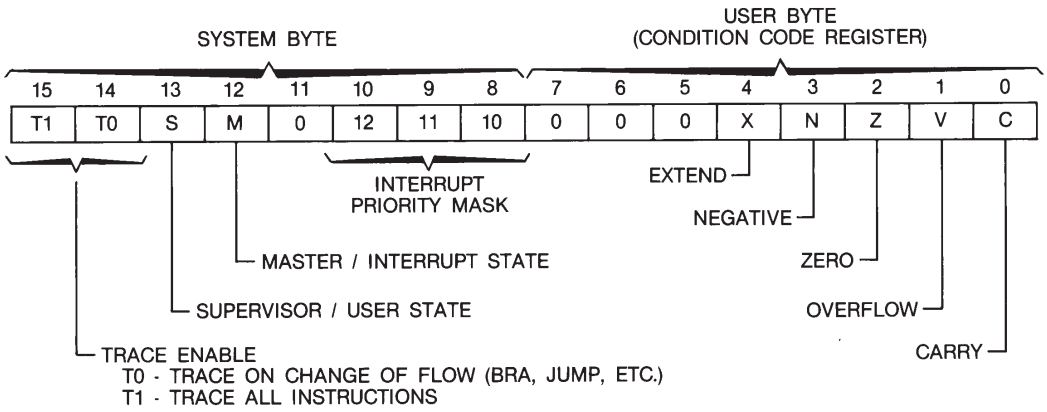


Figure 18 : Status register.

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 bits),
- Word Integers (16 bits),
- Long Word Integers (32 bits),
- Quad Word Integers (64 bits).

In additions, operations on other data types, such as memory addresses, status word data, etc..., are provided in the instruction set. The coprocessor mechanism allows direct support of floating-point data types with the TS 68881 and TS 68882 floating-point coprocessors, as well as specialized user-defined data types and functions.

The 18 addressing modes, shown in Table 8, include nine basic types :

- Register Direct,
- Register Indirect,
- Register Indirect with Index,
- Memory Indirect,
- Program Counter Indirect with Displacement,
- Program Counter Indirect with Index,
- Program Counter Memory Indirect,
- Absolute,
- Immediate.

The register indirect addressing modes support postincrement, predecrement, offset, and indexing. Programmers find these capabilities particularly useful for handling advanced data structures common to sophisticated applications and high level languages. The program counter relative mode also has index and offset capabilities ; programmers find that this addressing mode is required to support position-independent software. In addition to these addressing modes, the TS 68020 provides data operand sizing and scaling ; these features provide performance enhancements to the programmer.

Table 8 - TS 68020 addressing modes

Addressing Modes	Syntax
Register Direct	
Data Register Direct	Dn
Address Register Direct	An
Register Indirect	
Address Register Indirect	(An)
Address Register Indirect with Post Increment	(An) +
Address Register Indirect with Predecrement	− (An)
Address Register Indirect with Displacement	(d_{16} An)
Register Indirect with Index	
Address Register Indirect with Index (8-Bit Displacement)	(d_8, An, Xn)
Address Register Indirect with Index (Base Displacement)	(bd, An, Xn)
Memory Indirect	
Memory Indirect Post-Indexed	([bd, An], Xn, od)
Memory Indirect Pre-Indexed	([bd, An, Xn], od)
Program Counter Indirect with Displacement	(d_{16}, PC)
Program Counter Indirect with Index	
PC Indirect with Index (8-Bit Displacement)	d_8, PC, Xn)
PC Indirect with Index (Base Displacement)	(bd, PC, Xn)
Program Counter Memory Indirect	
PC Memory Indirect Post-Indexed	([bd, PC], Xn, od)
PC Memory Indirect Pre-Indexed	([bd, PC, Xn]), od)
Absolute	
Absolute Short	xxx.W
Absolute Long	xxx.L
Immediate	= (data)

Notes :

Dn = Data Register, D0-D7.

An = Address Register, A0-A7.

d_8, d_{16} = A twos-complement, or sign—extended displacement ; added as part of the effective calculation ; size is 8 (d_8) or 16 (d_{16}) bits ; when omitted assemblers use a value of zero.

Xn = Address or data register used as an index register ; form is Xn, SIZE*SCALE, where SIZE is .W or .L (indicates index register size) and SCALE is 1, 2, 4 or 8 (index register is multiplied by SCALE) ; use of SIZE and / or SCALE is optionnal.

bd = A two-complement base displacement ; when present, size can be 16 or 32 bits.

od = Outer displacement, added as part of effective address calculation after any memory indirection ; use is optional with a size of 16 or 32 bits.

PC = Program Counter.

(data) = Immediate value of 8, 16 or 32 bits.

( ) = Effective Address.

[ ] = Use as indirect address to long word address.

Instruction set overview

The TS 68020 instruction set is shown in Table 9. Special emphasis has been given to the instruction set's support of structured high-level languages and sophisticated operating systems. Each instruction, with few exceptions, operates on bytes, words, and long words and most instructions can use any of the 18 addressing modes. Many instruction extensions have been made on the TS 68020 to take advantage of the full 32-bit operation where, on the earlier 68000 Family members, only 8- and 16-bit values were used. The TS 68020 is upward source- and object-level code compatible with the family because it supports all of the instructions that previous family members offer. Additional instructions are now provided by the TS 68020 in support of its advanced features.

Table 9 - Instruction Set

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
ADDA	Add Address
ADDI	Add Immediate
ADDQ	Add Quick
ADDX	Add with Extend
AND	Logical AND
ANDI	Logical AND Immediate
ASL, ASR	Arithmetic Shift Left and Right
Bcc	Branch Conditionally
BCHG	Test Bit and Change
BCLR	Test Bit and Clear
BFCHG	Test Bit Field and Change
BFCLR	Test Bit Field and Clear
BFEXTS	Signed Bit Field Extract
BFEXTU	Unsigned Bit Field Extract
BFFFO	Bit Field Find First One
BFINS	Bit Field Insert
BFSET	Test Bit Field and Set
BFTST	Test Bit Field
BKPT	Breakpoint
BRA	Branch
BSET	Test Bit and Set
BSR	Branch to Subroutine
BTST	Test Bit
CALLM	Call Module
CAS	Compare and Swap Operands
CAS2	Compare and Swap Dual Operands
CHK	Check Register Against Bound
CHK2	Check Register Against Upper and Lower Bounds
CLR	Clear
CMP	Compare
CMPA	Compare Address
CMPI	Compare Immediate
CMPM	Compare Memory to Memory
CMP2	Compare Register Against Upper and Lower Bounds
DBcc	Test Condition, Decrement and Branch
DIVS, DIVSL	Signed Divide
DIVU, DIVUL	Unsigned Divide
EOR	Logical Exclusive OR
EORI	Logical Exclusive OR Immediate
EXG	Exchange Registers
EXT, EXTB	Sign Extend
ILLEGAL	Take Illegal Instruction Tape
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link and Allocate
LSL, LSR	Logical Shift Left and Right
MOVE	Move
MOVEA	Move Address
MOVE CCR	Move Condition Code Register
MOVE SR	Move Status Register
MOVE USP	Move User Stack Pointer
MOVEC	Move Control Register
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral
MOVEQ	Move Quck
MOVES	Move Alternate Address Sapce
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NEGX	Negate with Extend
NOP	No Operation
NOT	Logical Complement
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 from 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

Coprocessor Instructions

Mnemonic	Description
cpBCC	Branch Conditionally
cpDBcc	Test Coprocessor Condition, Decrement and Branch
cpGEN	Coprocessor General Instruction

Mnemonic	Description
cpRESTORE	Restore Internal State of Coprocessor
cpSAVE	Save Internal State of Coprocessor
cpScc	Set Conditionally
cpTRAPcc	Trap Conditionally

Bit field operations

The TS 68020 supports variable length bit field operations up to 32 bits. A bit field may start in any bit position and span any address boundary for the full length of the bit field, up to the 32 bit maximum. The bit field insert (BFINS) inserts a value into a field. Bit field extract unsigned (BFEXTU) and bit field extract signed (BFEXTS) extract a unsigned or signed value from the field. BFFFO finds the first bit in a bit field that is set. To complement the TS 68000 bit manipulation instruction, there are bit field change, clear, set, and test instructions (BFCHG, BFCLR, BFSET, BFTST). Using the on-chip barrel shifter, the bit and bit field instructions are very fast and particularly useful in applications using packed bits and bit fields, such as graphics and communications.

Binary coded decimal (BCD) support

The TS 68000 Family supports BCD operations including add, subtract, and negation. The TS 68020 adds the PACK and UNPACK operations for BCD conversions to and from binary form as well as other conversions, e.g., ASCII and EBCDIC. The PACK instruction reduces two bytes of data into a single byte while UNPACK reverses the operation.

Bounds checking

Previous 68000 Family members offer variable bounds checking only on the upper limit of the bound. The underlying assumption is that the lower bound is zero. This is expanded on the TS 68020 by providing two new instructions, CHK2 and CMP2. These instructions allow checking and comparing of both the upper and lower bounds. These instructions may be either signed or unsigned. The CMP2 instructions sets the condition codes upon completion while the CHK2 instruction, in addition to setting the condition codes, will take a system trap if either boundary condition is exceeded.

System traps

Three additions have been made to the system trap capabilities of the TS 68020. The current TRAPV (trap on overflow) instruction has been expanded to a TRAPcc format where any condition code is allowed to be the trapping condition. And, the TRAPcc instruction is expanded to optionally provide one or two additionnal words following the trap instruction so user-specified information may be presented to the trap handler. These additional words can be used when needed to provide simple error codes or debug information for interactive runtime debugging or post-mortem program dumps. Compilers may provide direction to run-time execution routines towards handling of specific conditions.

The breakpoint instruction, BKPT, is used to support the program breakpoint function for debug monitors and real-time in-circuit or hardware emulators, and the operation will be dependent on the actual system implementation. Execution of this instruction causes the TS 68020 to run a breakpoint acknowledge bus cycle, with a 3-bit breakpoint identifier placed on address lines A2, A3, and A4. This 3-bit identifier permits up to eight breakpoints to be easily differentiated. The normal response to the TS 68020 is an operation word (typically an instruction, originally replaced by the debugger with the breakpoint instruction) placed on the data lines by external debugger hardware and the breakpoint acknowledge cycle properly terminated. The TS 68020 then executes this operation word in place of the breakpoint instruction. The debugger hardware can count the number of executions of each breakpoint and halt execution after a pre-determined number of cycles.

Multi-processing

To further support multi-processing with the TS 68020, 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 varient 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 TS 68020 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 TS 68020 does not interpret 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 TS 68020 to perform the necessary actions to verify legitimate access to modules.

Virtual memory / machine concepts

The full addressing range of the TS 68020 is 4 gigabytes (4, 294, 967, 296). However, most TS 68020 systems implement a smaller physical memory. Nonetheless, by using virtual memory techniques, the system can be made to appear to have a full 4 gigabytes of physical memory available to each user program. These techniques have been used for many years in large mainframe computers and minicomputers. With the TS 68020 (as with the TS 68010 and TS 68012), 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 emulated system is called a virtual machine.

Virtual memory

The basic mechanism for supporting virtual memory is to provide a limited amount of high-speed physical memory that can be accessed directly by the processor while maintaining an image 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 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 TS 68020 uses instruction continuation to support virtual memory. In order for the TS 68020 to use instruction continuation, it stores its internal state on the supervisor stack when a bus cycle is terminated with a bus error signal. It then loads the program counter with the address of the virtual memory bus error handler from the exception vector table (entry number two) and resumes program execution at that new address. When the bus error exception handler routine has completed execution, an RTE instruction is executed which reloads the TS 68020 with the internal state stored on the stack, reruns the faulted bus cycle (when required), and continues the suspended instruction.

Instruction continuation is crucial to the support of virtual I/O devices in memory-mapped input / output systems. Since the registers of a virtual device may be simulatated in the memory map, an access to such a register will cause a fault and the function of the register can be emulated by software.

Virtual machine

A typical use for a virtual machine system is the development of software, such as an operating system, for a new machine also under development and not yet available for programming use. In such a system, a governing operating system emulates the hardware of the prototype system and allows the new operating system to be executed and debugged as though it were running on the new hardware. Since the new operating system is controlled by the governing operating system, it is executed at a lower privilege level than the governing operating system. Thus, any attempts by the new operating system to use virtual resources that are not physically present (and should be emulated) are trapped to the governing operating system and handled by its software. In the TS 68020, a virtual machine is fully supported by running the new operating system in the user mode. The governing operating system executes in the supervisor mode and any attempt by the new operating system to access supervisor resources or execute privileged instructions will cause a trap to the governing operating system.

Operand transfer mechanism

Though the TS 68020 has a full 32-bit data bus, it offers the ability to automatically and dynamically downsize its bus to 8 or 16 bits if peripheral devices are unable to accomodate the entire 32 bits. This feature allows the programmer the ability to write code that is not bus-width specific. For example, long word (32 bit) accesses to peripherals may be used in the code, yet the TS 68020 will transfer only the amount of data that the peripheral can manage. This feature allows the peripheral to define its port size as 8, 16, or 32 bits wide and the TS 68020 will dynamically size the data transfer accordingly, using multiple bus cycles when necessary. Hence, programmers are not required to program for each device port size or know the specific port size before coding ; hardware designers have flexibility to choose implementations independent of software prejudices.

This is accomplished through the use of the $\overline{\text{DSACK}}$ pins and occurs on a cycle-by-cycle basis. For example, if the processor is executing an instruction that requires the reading of a long word operand, it will attempt to read 32 bits during the first bus cycle to a long word address boundary. If the port responds that it is 32 bits wide, the TS 68020 latches all 32 bits of data and continues. If the port responds that it is 16 bits wide, the TS 68020 latches 16 valid bits of data an runs another cycle to obtain the other 16 bits of data. An 8-bit port is handled similarly but with four bus read cycles. Each port is fixed in assignment to particular sections of the data bus.

Justification of data on the bus is handled automatically by dynamic bus sizing. When reading 16-bit data from a 32 -bit port, the data may appear on the top or bottom half of the bus, depending on the address of the data. The TS 68020 determines which portion of the bus is needed to support the transfer and dynamically adjusts to read or write the data on those data lines.

The TS 68020 will always transfer the maximum amount of data on all bus cycles ; i.e., it always assumes the port is 32 bits wide when beginning the bus cycle. In addition, the TS 68020 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 TS 68020 aligned data requires multiple bus cycles, the TS 68020 automatically runs the minimum number of bus cycles.

The coprocessor concept

The coprocessor interface is a mechanism for extending the instruction set of the TS 68000 Family. Examples of these extensions are the addition of specialized data operands for the existing data types or, for the case of floating point, the inclusion of new data types and operations for them as implemented by the TS 68881 and TS 68882 floating-point coprocessors.

The programmer's model for the TS 68000 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 responsabilities since sequencing control is automatic and discrete.

The TS 68000 coprocessor interface is designed to extend the programmers model and it provides full support for the sequential, non-concurrent instruction execution model. Hence, instruction execution by the coprocessor is assumed to not overlap with instruction execution with the main microprocessor. Yet, the TS 68000 coprocessor interface does allow concurrent operation when concurrency can be properly accomodated. For example, the TS 68881 or TS 68882 floating-point coprocessor will allow the TS 68020 to proceed executing instructions while the coprocessor continues a floating-point operation, up to the point that the TS 68020 sends another request to the coprocessor. Adhering to the sequential execution model, the request to the coprocessor continues a floating-point operation, up to the coprocessor completes each TS 68881 and TS 68882 instruction before is starts the next, and the TS 68020 is allowed to proceed as it can in a concurrent fashion.

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

The communication protocol between the main processor and the coprocessor necessary to execute a coprocessor instruction is based on a group of coprocessor interface registers (Table 10) which are defined for the TS 68000 Family coprocessor interface. The TS 68020 hardware uses standard TS 68000 asynchronous bus cycles to access the registers. Thus, the coprocessor doesn't require a special bus hardware ; the bus interface implemented by a coprocessor for its interface register set must only satisfy the TS 68020 address, data, and control signal timing to guarantee proper communication with the main processor. The TS 68020 implements the communication protocol with all coprocessors in hardware (and microcode) and handles all operations automatically so the programmer is only concerned with the instructions and data types provided by the coprocessor as extensions to the TS 68020 instruction set and data types.

Other microprocessors in the TS 68000 Family can operate any TS 68000 coprocessor even though they may not have the hardware implementation of the coprocessor interface as does the TS 68020. Since the coprocessor is operated through the coprocessor interface registers which are accessed via normal asynchronous bus cycles, the coprocessor may be used as a peripheral device. Software easily emulates the communication protocol by addressing the coprocessor interface registers appropriately and passing the necessary commands and operands required by the coprocessor.

The coprocessor interface registers are implemented by the coprocessor in addition to those registers implemented as extensions to the TS 68020 programmer's model. For example, the TS 68881 implements the coprocessor 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 TS 68881 programmer.

Up to eight coprocessors are supported in a single system with a system-unique coprocessor identifier encoded in the coprocessor instruction. When accessing a coprocessor, the TS 68020 executes standard read and write bus cycle in CPU address

Table 10 · Coprocessor interface registers

Register	Function	R / W
Response	Requests Action from CPU	R
Control	CPU Direct Control	W
Save	Initiate Save of Internal State	R
Restore	Initiate Restore of Internal State	R / W
Operation Word	Current Coprocessor Instruction	W
Command Word	Coprocessor 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 Coprocessor Instruction	R / W
Operand Address	Pointer to Coprocessor Operand	R /W

Table 11 · Coprocessor primitives

Processor Synchronization Busy with Current Instruction Proceed with Next Instruction, If No Trace Service Interrupts and Re-query, If Trace Enable Proceed with Execution, Condition True / False
Instruction Manipulation Transfer Operation Word Transfer Words from Instruction Stream
Exception Handling Take Privilege Violation if S Bit Not Set Take Pre-Instruction Exception Take Mid-Instruction Exception Take Post-Instruction Exception
General Operand Transfer Evaluate and Pass (ea) Evaluate (ea) and Transfer Data Write to Previously Evaluated (ea) Take Address and Transfer Data Transfer to / from Top of Stack
Register Transfer Transfer CPU Register Transfer CPU Control Register Transfer Multiple CPU Registers Transfer Multiple Coprocessor Registers Transfer CPU SR and / or ScanPC

space, as encoded by the function codes, and places the coprocessor identifier on the address bus to be used by chip-select logic to select the particular coprocessor. Since standard bus cycle are used to access the coprocessor, the coprocessor may be located according to system design requirements, whether it be located on the microprocessor local bus, on another board on the system bus, or any other place where the chip-select and coprocessor protocol using standard TS 68000 bus cycles can be supported.

Coprocessor protocol

Interprocessor transfers are all initiated by the main processor during coprocessor instruction execution. During the processing of a coprocessor instruction, the main processor transfers instruction information and data to the associated coprocessor, and receives data, requests, and status information from the coprocessor. These transfers are all based on the TS 68000 bus cycles.

The typical coprocessor protocol which the main processor follows is :

a) The main processor initiates the communications by writing command information to a location in the coprocessor interface.

b) The main processor reads the coprocessor response to that information.

   1) The response may indicate that the coprocessor is busy, and the main processor should again query the coprocessor. This allows the main processor and coprocessor to synchronize their concurrent operations.

   2) The response may indicate some exception condition ; the main processor acknowledges the exception and begins exception processing.

   3) The response may indicate that the coprocessor needs the main processor to perform some service such as transferring data to or from the coprocessor. The coprocessor may also request that the main processor query the coprocessor again after the service is complete.

   4) The response may indicate that the main processor is not needed for further processing of the instruction. The communication is terminated, and the main processor is free to begin execution of the next instruction. At this point in the coprocessor protocol, as the main processor continues to execute the instruction stream, the main processor may operate concurrently with the coprocessor.

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

Each coprocessor instruction type has specific requirements based on this simplified protocol. The coprocessor interface may use as many extension words as required to implement a coprocessor instruction.

Primitives / response

The response register is the means by which the coprocessor communicates service requests to the main processor. The content of the coprocessor response register is a primitive instruction to the main processor which is read during coprocessor communication by the main processor. The main processor «executes» this primitive, thereby providing the services required by the coprocessor. Table 11 summarizes the coprocessor primitives that the TS 68020 accepts.

Exceptions

Kinds of exceptions

Exceptions 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 instruction 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 proccesing 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 coprocessor detected exceptions, the vector number is included in the coprocessor exception primitive response. For all other exceptions, internal logic provides the vector number. This vector number is then used to generated 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 exeption 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 TS 68020 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 related 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 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 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 TS 68020 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.

TS 68020 cache goals

There where two primary goals for the TS 68020 microprocessor cache. The first design goal was to reduce the processor external bus activity. In a given TS 68000 system, the TS 68000 processor will use approximately 80 to 90 percent (for greater) of the available bus bandwidth. This is due to its extremely efficient prefetching algorithm and the overall speed of its internal architecture design. Thus, in an TS 68000 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 TS 68020 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 TS 68020 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 TS 68020 is gained in two ways. First, the TS 68020 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.

Second, and probably the most important benefit of the cache, is that it allows instruction stream fetches and operand accesses to proceed in parallel. For example, if the TS 68020 requires both an instruction stream access and an operand access, and the instruction is resident in the cache, the operand access will proceed unimpeded rather than being queued behind the instruction fetch. Similarly, the TS 68020 is fully capable of executing several internal instructions (instructions that do not require the bus) while completing an operand access for another instruction.

The TS 68020 instruction cache is a 256-byte direct mapped cache organized as 64 long word entries. Each cache entry consists of a tag field made up of the upper 24 address bits, the FC2 (user / supervisor) value, one valid bit, and 32 bits of instruction data (Figure 19).

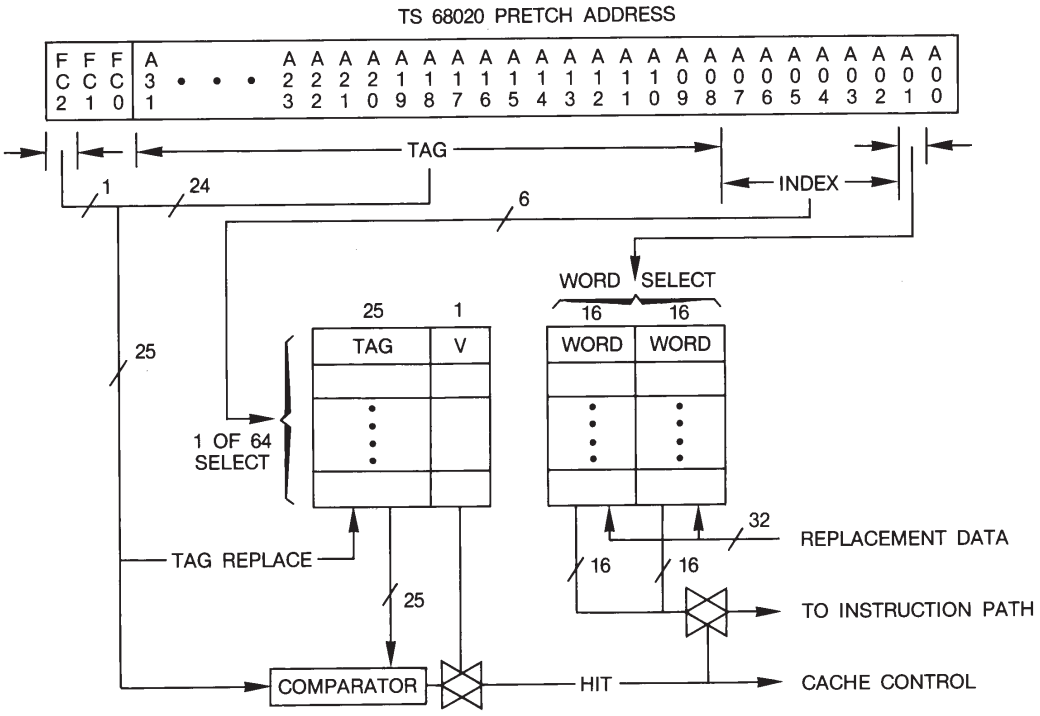


Figure 19 : TS 68020 on-chip cache organization.

The TS 68020 employs a 32 bit data bus and fetches instructions on long word address boundaries. Hence, each 32-bit instruction fetch brings in two 16-bit instruction words which are then written into the on-chip cache. When the cache is enabled, the subsequent prefetch will find the next 16-bit instruction word is already present in the cache and the related bus cycle is saved. If the cache were not enabled, the subsequent prefetch will find the bus controller still holds the full 32 bits and can satisfy the prefetch and again save the related bus cycle. So, even when the on-chip instruction cache is not enabled, the bus controller provides an instruction «cache hit» rate up to 50 %.

7 · PREPARATION FOR DELIVERY

7.1 · Packaging

Microcircuit are prepared for delivery in accordance with MIL-M-38510.

7.2 · Certificate of compliance

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

8 · 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.

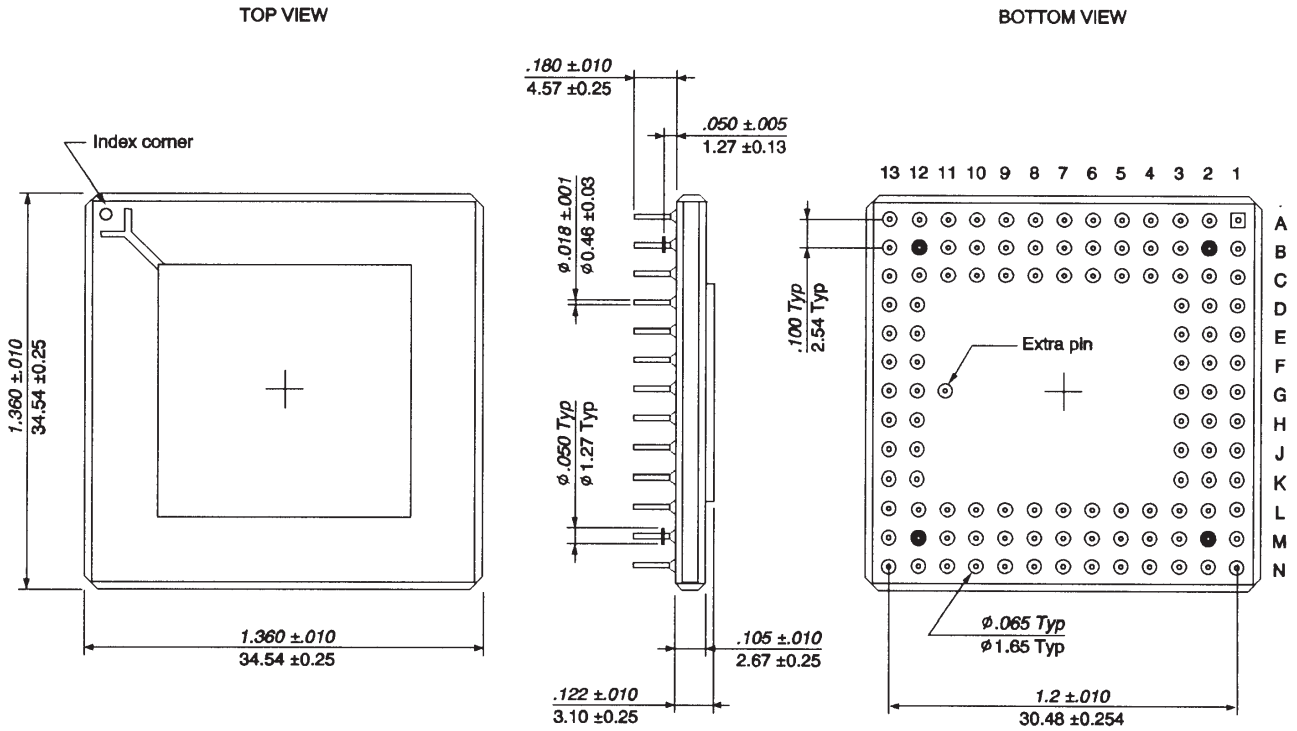
e) Avoid use of plastic, rubber, or silk in MOS areas.

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

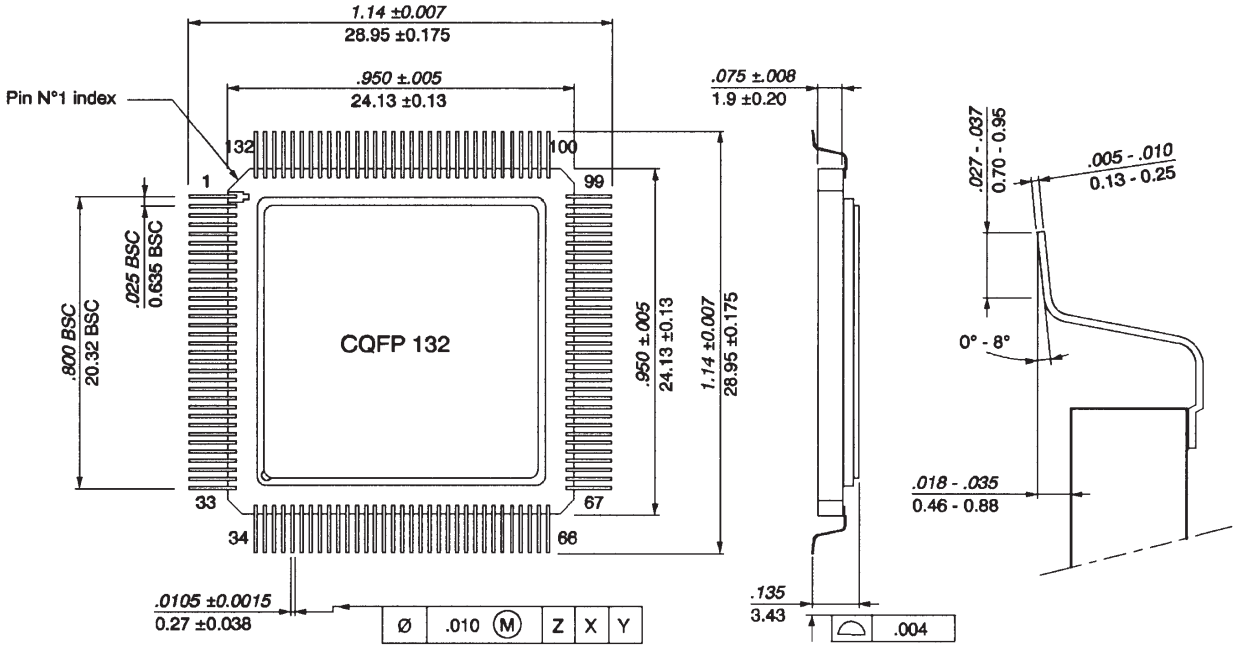
9 - PACKAGE MECHANICAL DATA

9.1 - 114 pins - Ceramic Pin Grid Array

Conform to MIL-M-38510, appendix C, P-AE outline.



9.2 - 132 pins - Ceramic Quad Flat Pack



9.3 - Mass

PGA 114 - 6 grams typically

CQFP 132 - 14 grams typically

10 - TERMINAL CONNECTIONS

10.1 - 114 pins - Ceramic Pin Grid Array

See Figure 2.1 page 4.

10.2 - 132 pins - Ceramic Quad Flat Pack

See Figure 2.2 page 4.

11 - ORDERING INFORMATION

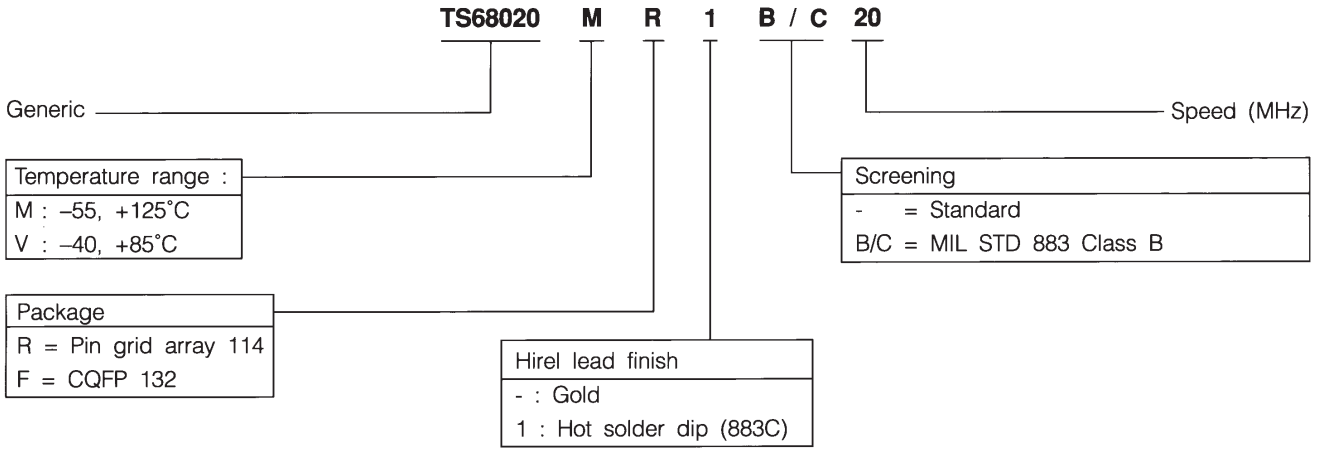
11.1 - Hi-REL product

Commercial TCS Part-Number (see Note)	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	5962-86032XA
TS68020DESC03XA	DESC	PGA 114 / tin	– 55 / + 125	20	5962-86033XA
TS68020DESC04XA	DESC	PGA 114 / tin	– 55 / + 125	25	5962-86034XA
TS68020DESC02XC	DESC	PGA 114	– 55 / + 125	16.67	5962-86032XC
TS68020DESC03XC	DESC	PGA 114	– 55 / + 125	20	5962-86033XC
TS68020DESC04XC	DESC	PGA 114	– 55 / + 125	25	5962-86034XC
TS68020DESC02YA	DESC	CQFP 132 / tin	– 55 / + 125	16.67	5962-86032YA
TS68020DESC03YA	DESC	CQFP 132 / tin	– 55 / + 125	20	5962-86033YA
TS60020DESC04YA	DESC	CQFP 132 / tin	– 55 / + 125	25	5962-86034YA
TS68020DESC02YC	DESC	CQFP 132	– 55 / + 125	16.67	5962-86032YC
TS68020DESC03YC	DESC	CQFP 132	– 55 / + 125	20	5962-86033YC
TS60020DESC04YC	DESC	CQFP 132	– 55 / + 125	25	5962-86034YC
Note : THOMSON-CSF SEMICONDUCTEURS SPECIFIQUES.					

11.2 · Standard product

Commercial TCS Part-Number (see Note)	Norms	Package	Temperature range T_C (°C)	Frequency (MHz)	Drawing number
TS68020VR16	TCS Standard	PGA 114	– 40 / + 85	16.67	Internal
TS68020VR20	TCS Standard	PGA 114	– 40 / + 85	20	Internal
TS68020VR25	TCS Standard	PGA 114	– 40 / + 85	25	Internal
TS68020MR16	TCS Standard	PGA 114	– 55 / + 125	16.67	Internal
TS68020MR20	TCS Standard	PGA 114	– 55 / + 125	20	Internal
TS68020MR25	TCS Standard	PGA 114	– 55 / + 125	25	Internal
TS68020VF16	TCS Standard	CQFP 132	– 40 / + 85	16.67	Internal
TS68020VF20	TCS Standard	CQFP 132	– 40 / + 85	20	Internal
TS68020VF25	TCS Standard	CQFP 132	– 40 / + 85	25	Internal
TS68020MF16	TCS Standard	CQFP 132	– 55 / + 125	16.67	Internal
TS68020MF20	TCS Standard	CQFP 132	– 55 / + 125	20	Internal
TS68020MF25	TCS Standard	CQFP 132	– 55 / + 125	25	Internal

Note : THOMSON-CSF SEMICONDUCTEURS SPECIFIQUES.



This product is manufactured by THOMSON-CSF SEMICONDUCTEURS SPECIFIQUES - 38521 SAINT-EGREVE / FRANCE.

For further information please contact : THOMSON-CSF SEMICONDUCTEURS SPECIFIQUES - Route Départementale 128 - B.P. 46 - 91401 ORSAY Cedex / FRANCE - Tél. : (33)(0) 1.69.33.03.24 / Téléfax : (33)(0) 1.69.33.03.21.
Internet : http://www.atmel-grenoble.com

ORDER CODE : DSTS68020/0500 Imprimé et Réalisé en France par Graphic Express - Tél. : 01.46.55.27.24 - 10780 - 05/00