REFERENCE GUIDE

SYSTEMS

S-3260

Execution Times
of
TEKTEST III Statements

TEKTRONIX

August 1977

INTRODUCTION

This *Reference Guide* lists the execution times for TEKTEST™ III statements. Execution time depends upon the nature of the statement itself (basic execution time), the complexity of expressions and pinlists used within the statement (expression and pinlist evaluation times), and the numeric value of the data processed. Generally, execution time is the sum of the basic execution time, the expression evaluation time, and the pinlist evaluation time. That is,

statement execution time = basic time + expression time + pinlist time

Execution times given are typical and may vary with the data processed, memory speed, processor speed, and UNIBUS length. Timings do not include the additional overhead which results from using the TRACE option.

Listed under "Expression Evaluation Times" are the execution times for operators, system functions, and reserved variables (for example, multiplication, SQRT, and CURRENT). To determine total expression execution time, take the sum of the times for all operators, functions, and reserved variables in the expression.

Listed under "Pinlist Evaluation Times" are the execution times for evaluating indexed pinlists.

Listed in alphabetical order under "Basic Execution Times" are the execution times for TEKTEST III statements. Not included are nonexecutable statements, such as ARRAY, and user-time-dependent statements, such as ACCEPT.

Execution times listed are guidelines, not system specifications. Because software and hardware improvement is a continuing process, Tektronix reserves the right to change without prior notice any execution time listed in this Guide.

ITEM	*TIME*
Mathematical Operators	
+ (Addition)	95 μs
/ (Division)	120 μs
** (Exponentiation)	1720 μs
* (Multiplication)	110 μs
– (Subtraction)	95 μs
Logical Operators	
AND (Operator)	60 μs
EQ	115 μs
GE	115 μs
GT	125 μs
LE	120 μs
LT	120 μs
NE	115 μs
NOT (Operator)	30 μs
OR (Operator)	55 μs

ITEM	*TIME*
System Functions	
ABS	40 μs
AND (Function)	245 μs
ATAN	570 μs (Minimum) 920 μs (Maximum)
COS	800 μs (Minimum) 890 μs (Maximum)
EXP	720 μs
GETBUS	160 μs
INT	75 μs (Minimum) 110 μs (Maximum)
LOG (Base e)	785 μs
LOG10	800 μs
OR (Function)	245 μs
SETBUS	245 μs
SIN	780 μs (Minimum) 870 μs (Maximum)
SQRT	335 μs
XOR (Function)	240 μs

ITEM	*TIME*
Using Array Variables and Common Variables	
ARRAY Variables	
Singly Subscripted	150 μs
Doubly Subscripted	240 μs
COMMON Variable	10 μs
IARRAY Variable	235 μs
Using Reserved Variables	
ADVANCE	30 μs
AUTOV	
100 mV Range	4.2 ms
1 V Range	0.9 ms
10 V Range	0.9 ms
100 V Range	1.5 ms
(Take the sum of the times for all ranges used. Then add 1 ms for each range change when using automatic ranging.)	
CLOCK	95 μs

(Continued)

ITEM	*TIME*
Using Reserved Variables (Continued)	
CURRENT	
100 nA Range (from driver)	25.2 ms
1 μA Range (from driver)	9.2 ms
10 μA Range (from driver)	6.2 ms
100 μA Range (from driver)	0.7 ms
1 mA Range (from driver)	0.2 ms
10 mA Range (from driver)	0.2 ms
100 mA Range (from driver)	0.2 ms
100 nA Range (not driver)	25.2 ms
1 μA Range (not driver)	8.2 ms
10 μA Range (not driver)	1.7 ms
100 μA Range (not driver)	0.5 ms
1 mA Range (not driver)	0.2 ms
10 mA Range (not driver)	0.4 ms
100 mA Range (not driver)	1.7 ms
450 mA Range (not driver)	10.2 ms
ERROR	20 μs
INDEX	95 μs
TIME	80 μs
VOLTAGE	
100 mV Range	4.2 ms
1 V Range	0.9 ms
10 V Range	0.9 ms
100 V Range	1.5 ms

ITEM	*TIME*
Singly Indexed Pinlist	320 μs
Doubly Indexed Pinlist (Add 38 μs for each pin in the pinlist.)	375 μs

STATEMENT NAME AND COMMENTS	*TIME*
BURST ON	
(1) If any 2941 values have changed since the last MOVE or BURST ON and	
(a) cycle is $< 16\ \mu s$:	365 μs
(b) cycle is $> 16\ \mu s$:	405 μs
(2) No 2941 values have been changed:	50 μs
BURST OFF	40 μs
CALL {**linenumber** (Simple CALL); **(expression) linenumber, . . . , linenumber** (Computed CALL)}	68 μs
(1) Transfer of program flow occurs:	165 μs
(2) Expression too large. No transfer of program flow occurs:	130 μs
CLEAR ERROR	40 μs

STATEMENT NAME AND COMMENTS	*TIME*
COMPARE pins WITH	
ONE ZERO PATTERN TOGGLE ONE AND ZERO TOGGLE ZERO AND ONE ALTERNATE PATTERN AND ONE ALTERNATE PATTERN AND ZERO	And the number of groups (W,X,Y,Z) is: 1 → 150 μs; 2 → 175 μs; 3 → 195 μs; 4 → 220 μs
variable TOGGLE variable ALTERNATE PATTERN AND variable	Add 50 μs for each "1" bit, 15 μs for each imbedded "0" bit, and 25 μs for each group. Do not add any pinlist time. → 475 μs

STATEMENT NAME AND COMMENTS		TIME
CONNECT . . . TO		
REGISTER, SECTOR, DBUS, EBUS, COMPUTER	And the number of groups (W,X,Y,Z) is: 1	68 μs
	2	75 μs
	3	82 μs
	4	88 μs
SCOPE1, SCOPE2, AUX1, AUX2	Without the ;ND (no delay) element:	126 ms
	With the ;ND (no delay) element:	0.19 ms
COMPARATOR, DRIVER, LOAD1, LOAD2, PHASE, DATAPHASE, GROUND	Without the ;ND (no delay) element, and the number of groups (W,X,Y,Z) is: 1	1.15 ms
	2	1.16 ms
	3	1.18 ms
	4	1.19 ms

(Continued)

STATEMENT NAME AND COMMENTS		TIME
(Continued)		
CONNECT . . . TO		
COMPARATOR DRIVER LOAD1 LOAD2 PHASE DATAPHASE GROUND	With the ;ND (no delay) element, and the number of groups (W,X,Y,Z) is:	80 μs 90 μs 100 μs 110 μs
COMMON	Without the ;ND ı) element, and the number o` rou (W,X,Y,Z) is:	2.25 ms 2.26 ms 2.27 ms 2.28 ms
COMMON	With the ;ND (no and the number ot X ,Z) is:	1.20 ms 1.21 ms 1.22 ms 1.23 ms

STATEMENT NAME AND COMMENTS	*TIME*
CYCLE = period	
(1) If *period* is a constant:	115 μs
(2) If *period* is not a constant:	280 μs
DATAPHASE = start FOR duration	
(1) If both *start* and *duration* are constants:	155 μs
(2) If either *start* or *duration* is not a constant:	480 μs
DISCONNECT . . . FROM	
{ SCOPE1 / SCOPE2 / AUX1 / AUX2 } Without the ;ND (no delay) element:	1.17 ms
{ SCOPE1 / SCOPE2 / AUX1 / AUX2 } With the ;ND (no delay) element:	110 μs

(Continued)

STATEMENT NAME AND COMMENTS		TIME
(Continued)		
DISCONNECT . . . FROM		
COMPARATOR DRIVER LOAD1 LOAD2 GROUND	Without the ;ND (no delay) element, and the number of groups (W,X,Y,Z) is: 1	1.13 ms
	2	1.14 ms
	3	1.16 ms
	4	1.17 ms
COMPARATOR DRIVER LOAD1 LOAD2 GROUND	With the ;ND (no delay) element, and the number of groups (W,X,Y,Z) is: 1	68 μs
	2	75 μs
	3	82 μs
	4	88 μs
COMMON ALL	Without the ;ND (no delay) element, and the number of groups (W,X,Y,Z) is: 1	2.25 ms
	2	2.26 ms
	3	2.27 ms
	4	2.28 ms

(Continued)

STATEMENT NAME AND COMMENTS	TIME
(Continued)	
DISCONNECT . . . FROM	
{COMMON / ALL} With the ;ND (no delay) element, and the number of groups (W,X,Y,Z) is: {1 / 2 / 3 / 4}	1.20 ms 1.21 ms 1.22 ms 1.23 ms
DISPLAY	290 μs
Equal (=)	85 μs
FORCE (1) Without RZ and INVERT. Timings are the same as for **COMPARE** (see above). (2) With RZ or INVERT. Timings are the same as for **FORCE,COMPARE** (see below).	

STATEMENT NAME AND COMMENTS	*TIME*
FORCE,COMPARE	
ONE, ZERO, PATTERN, TOGGLE ONE AND ZERO, TOGGLE ZERO AND ONE, ALTERNATE PATTERN AND ONE, ALTERNATE PATTERN AND ZERO — And the number of groups (W,X,Y,Z) is: 1	180 μs
2	212 μs
3	239 μs
4	272 μs
variable, TOGGLE variable, ALTERNATE PATTERN AND variable — Add 50 μs for each "1" bit, 15 μs for each imbedded "0" bit, and 25 μs for each group. Do not add any pinlist time.	475 μs
GO TO linenumber (Simple GO TO)	35 μs
GO TO (expression) linenumber, . . . ,linenumber (Computed GO TO)	
(1) Transfer of program flow occurs:	155 μs
(2) Expression too large. No transfer of program flow occurs:	135 μs

STATEMENT NAME AND COMMENTS	*TIME*
HICOMPARE { **= voltage AT range ON pins** / **FROM start FOR duration** }	**VS1** time + 4.4 ms
(1) If both *start* and *duration* are constants:	155 μs
(2) If either *start* or *duration* is not a constant:	480 μs
HIDRIVE = voltage ON pins	**VS2** time + 4.4 ms
IF { **(expression)** / **(expression$_1$,expression$_2$,expression$_3$) linenumber,linenumber,linenumber** }	65 μs
(1) *expression*$_2$ is less than *expression*$_1$	140 μs
(2) *expression*$_2$ is greater than *expression*$_3$	185 μs
(3) *expression*$_2$ is within limits set by *expression*$_1$ and *expression*$_3$	180 μs
INHIBIT Timings are the same as for **COMPARE** (see above).	
INITIALIZE	14 ms

STATEMENT NAME AND COMMENTS		*TIME*		
IS1 = current, limit		*Both current and limit are same as before.*	*Power supply polarity is same as before.*	*Power supply polarity is changed.*
	Both current and limit are constants.	0.1 ms	2.65 ms	3.45 ms
	Either current or limit is not a constant.	0.56 ms	3.15 ms	3.95 ms
IS2 = current, limit		*Both current and limit are same as before.*	*Power supply polarity is same as before.*	*Power supply polarity is changed.*
	Both current and limit are constants.	0.1 ms	2.65 ms	2.95 ms
	Either current or limit is not a constant.	0.56 ms	3.15 ms	3.45 ms

STATEMENT NAME AND COMMENTS		*TIME*
LOAD CORE	For each 16-bit word, add 1.6 μs to the time shown. The number of 16-bit words is equal to the number of rows to be loaded times the number of groups (W,X,Y, or Z) used. Multiply this total by 2 if you are using FC or FI,CM. Multiply the total by 4 if you are using FICM.	410 μs
LOAD DISK	Use this formula to calculate the time: **40 ms(INT((number of 16 bit words − 1)/256) − 1) + latency + seektime** Latency may be as great as 40 ms. Seektime will usually be zero, but may be as great as 100 ms.	
LOCOMPARE { **= voltage AT range ON pins** **FROM start FOR duration** }		**VS1** time + 4.4 ms
	(1) If both *start* and *duration* are constants:	155 μs
	(2) If either *start* or *duration* is not a constant:	480 μs

STATEMENT NAME AND COMMENTS		*TIME*
LODRIVE = voltage ON pins		**VS2** time + 4.4 ms
LOOP	Each LOOP statement creates code at two places: the start of the loop and after the last statement in the loop. The code at the start of the loop is only executed once – when the loop is entered. The code at the end of the loop is executed repeatedly until program flow transfers out of the loop.	
	Time to enter the loop if all loop values are constants:	165 μs
	Time to enter the loop if any loop value is not a constant:	420 μs
	Time for execution of the code at the end of the loop each time the loop is iterated:	150 μs
MASK	Timings are the same as for **COMPARE** (see above).	

STATEMENT NAME AND COMMENTS		TIME		
MOVE	*(Δ t_0 is the time between two consecutive t_0's.)*	*Cycle & phases not changed since last MOVE*	*Changed Δ $t_0 < 16$ μs*	*Changed Δ $t_0 > 16$ μs*
	Simple MOVE REGISTER	260 μs to 275 μs	475 μs	625 μs
	MOVE with TIMEOUT clause	385 μs to 520 μs	600 μs to 750 μs	750 μs to 880 μs
	MOVE with NO CONNECTS clause	230 μs to 275 μs	460 μs	605 μs
	MOVE R2942	390 μs to 500 μs	600 μs to 710 μs	750 μs to 850 μs
	Simple MOVE CORE Δ $t_0 < 10$ μs	430 μs	655 μs	
	Simple MOVE CORE Δ $t_0 \geq 10$ μs	485 μs to 635 μs	525 μs to 630 μs	885 μs to 985 μs

(Continued)

STATEMENT NAME AND COMMENTS			*TIME*
(Continued)			
MOVE · *(Δ t_O is the time between two consecutive t_O's.)*	*Cycle & phases not changed since last MOVE*	*Changed Δ t_O < 16 μs*	*Changed Δ t_O > 16 μs*
MOVE CORE with NO CONNECTS clause Δ t_O < 10 μs	400 μs	625 μs	
MOVE CORE with NO CONNECTS clause Δ t_O ⩾ 10 μs	470 μs to 620 μs	510 μs to 615 μs	900 μs to 965 μs
To the time in the table add the following:			
If START is a variable:			90 μs
If STOP is a variable:			90 μs
If APPEND is a variable:			90 μs
If the number of groups (W,X,Y,Z) is: 2			15 μs
3			30 μs
4			45 μs

(Continued)

STATEMENT NAME AND COMMENTS		TIME
(Continued)		
MOVE	If CHAIN(2):	95 μs
	If CHAIN(3):	110 μs
	If CHAIN(4):	125 μs
	If CHAIN(5) or more:	140 μs
	If MOVE REGISTER,CHAIN(2),AND SAVE ERRORS:	30 μs
	If MOVE REGISTER,CHAIN(3),AND SAVE ERRORS:	30 μs
	If MOVE REGISTER,CHAIN(4),AND SAVE ERRORS:	15 μs
	In addition, add the transfer time.	
	Transfer time for MOVE REGISTER: number of rows times the cycle rate.	
	Transfer time for MOVE CORE: number of rows times the cycle rate, or 1.6 μs times the number of 16-bit words. Compute the transfer time both ways and use the larger of the two numbers.	
PHASE1 through PHASE7	(1) If both *start* and *duration* are constants:	155 μs
	(2) If either *start* or *duration* is not a constant:	480 μs

STATEMENT NAME AND COMMENTS			*TIME*
RETURN			40 μs
RUN	Use this formula to compute the time:	Seektime:	50 to 360 ms
	seektime + latency + transfer time	Latency:	80 to 200 ms
	Typical RUN time is about 250 ms.	Transfer Time:	13 μs/ word
SETUP			
	TO MEASURE CURRENT FROM		
	DRIVER (10 mA to 99.9 mA)		5.65 ms
	DRIVER (not 10 mA to 99.9 mA)		3.65 ms
	VS3 or VS4 (10 mA to 99.9 mA)		**VS3** or **VS4** time + 5.4 ms
	VS3 or VS4 (not 10 mA to 99.9 mA)		**VS3** or **VS4** time + 3.4 ms

(Continued)

STATEMENT NAME AND COMMENTS	*TIME*
(Continued)	
SETUP	
TO MEASURE VOLTAGE	
Single pin	1.78 ms
Single pin AT RANGE	1.87 ms
Two pins AT RANGE	2.0 ms
TO FORCE CURRENT	
Positive current	3.8 ms
Negative current	4.6 ms
TO FORCE VOLTAGE	**VS3** or **VS4** time + 1.75 ms
TO MEASURE TIME	
ON pin TO pin	1.8 ms
FROM REFERENCE	1.65 ms

STATEMENT NAME AND COMMENTS		*TIME*
SORT		320 μs to 370 μs
TRACE	**ON**	30 μs
	OFF	30 μs
TRIGGER		125 μs
TSTRIG		30 μs
UNDERSOCKET		125 μs
UNSET	TO MEASURE VOLTAGE	
	Single pin	1.89 ms
	Two pins	1.95 ms
	TO MEASURE CURRENT	3.4 ms
	TO MEASURE TIME	1.42 ms
	TO FORCE VOLTAGE	1.36 ms

(Continued)

STATEMENT NAME AND COMMENTS	*TIME*
(Continued) **UNSET** TO FORCE CURRENT If IS1 was positive: if IS1 was negative:	 3.8 ms 2.0 ms

{VS1, VS2, VS3, VS4} = voltage, limit

	Both voltage and limit are same as before.	*Power supply polarity is same as before.*	*Power supply polarity is changed.*
Both voltage and limit are constants.	0.1 ms	2.65 ms	4.66 ms
Either voltage or limit is not a constant.	0.66 ms	3.23 ms	5.24 ms

STATEMENT NAME AND COMMENTS		*TIME*
WAIT	0 to 100 μs: Above 100 μs:	110 μs Programmed value
WHEN ERROR		55 μs
WHEN ERROR OR OVERFLOW		55 μs
WHEN OVERFLOW		55 μs
WHEN PASS		55 μs

Notes

Notes

Notes

Notes