

Q: How does the Pattern Generator RPEAT function work?

Repeating a Vector Using RPEAT

When you need to do pattern matching, or remain on a vector until a failure occurs, HiLevel has the perfect command for the Pattern Generator: RPEAT. This command tells the HiLevel high-speed command processor in the system to remain at this vector until either A) a failure occurs, or B) a given number of test rate clock cycles have occurred. If a failure occurs, the test vector execution drops off of the RPEAT function and goes to the next vector, which would be a Conditional Jump (CJMP) to a point that is two vectors later. See the example below:

Vectors: C:\EtsUsb\74F161.trn		
Vector Address (Dec)	Program	C CC T QQQQ L DDDD EE P M C 3210 K 3210 PT E R
0	NOOP	L LLLL 1 0000 11 1 0
1	NOOP	L LLLH 1 0000 11 1 1
2	NOOP	L LLHL 1 0000 11 1 1
3	RPEAT 00100	L LLHH 1 0000 11 1 1
4	CJMP 00006	L LHLL 1 0000 11 1 1
5	JMP 10000	L LHLH 1 0000 11 1 1
6	NOOP	L LHHL 1 0000 11 1 1
7	NOOP	L LHHH 1 0000 11 1 1

The RPEAT command includes a numerical value. In the case of the example, it is set to a value of 100. This value is similar a “time out”; if no failure is detected after 100 tester clock cycles, vector execution continues. Since a failure did not occur, the CJMP condition will not be true and the JMP at vector 5 will be executed so that you jump immediately to an error trap at vector 10,000. This vector 10,000 may contain a JMP 10000 (Jump to self) so that we know that no failure or pattern match occurred. (Vector 10,000 would not be part of our normal vectors, and would only be executed by “unsatisfied” RPEAT commands.)

In the case of a failure being detected during RPEAT, vector execution would continue and the tester Condition latch would be True, so the CJMP would be executed. This causes the vector sequence to jump *over* the JMP 10000 at vector 5 and continue executing normally. Additional RPEAT functions can appear later in the same vector set if desired.

The maximum value for RPEAT is 64K (65,535 decimal).