[8 BIT TIMER - TMR] TEST PLAN

[Clock + Reset] Tests								
No	**Test name**	**Type**	**Requirement**	**Description**	**Status**	**Start Date**	**End Date**	**Priority**

[Register] Tests								
No	**Test name**	**Type**	**Requirement**	**Description**	**Status**	**Start Date**	**End Date**	**Priority**
1	**tdr**_test.v	**Register**	*read write tdr*	*1. read tdr -> check default value 2. write random value to tdr (address = 0) 3. read tdr and compare written value 4. repeat 20 times*				
2	**tcr**_test.v		*read write tcr*	*1. read tcr -> check default value 2. write random value to tcr (address = 1) 3. read tcr and compare written value with mask = 1011_0011 4. repeat 20 times*				
3	**tsr**_test.v		*read write tsr*	*1. read tsr -> check default value 2. write random value to tsr (address = 2) 3. read tsr and compare 0 4. repeat 20 times*				
4	null_address.v		*read write non-existed address*	*-Write a random value to a random address. -Check if PSLVERR is triggered. If the address is not TDR (0), TCR (1), or TSR (3), display the message "NULL-ADDRESS"; otherwise, compare the read value with the written value. -Repeat 20 times.*				
5	mixed_address.v		*read write all of address address = random wdata = random*	*-Write a random value to a random address. -Check if PSLVERR is triggered. If the address is not TDR (0), TCR (1), or TSR (3), display the message "NULL-ADDRESS"; otherwise, compare the read value with the written value. -Repeat 20 times.*				

[Functionality] Tests								
No	**Test name**	**Type**	**Requirement**	**Description**	**Status**	**Start Date**	**End Date**	**Priority**
6	countup_forkjoin_pclk2.v	**Count Up**	*-Enter a random number less than 255. -Thread 1: Counts up with an internal clock of pclk × 2, starting from the number above. If the Thread 1 count-up exceeds 255, it checks if the overflow status is triggered. If so, it displays a "pass" message (normal operation); otherwise, it displays a "faulty" message. -Thread 2 runs counting-up in parallel with Thread 1, but Thread 2's job duration is shorter than Thread 1's (about 2/3 of Thread 1's duration). Thread 2 eventually stops its counting job (while Thread 1 is still running) and checks if an overflow occurred. If overflow occurs, it displays a "faulty" message; otherwise, it displays a "pass" (normal operation).*	*-Write a random value to TDR (address = 0). -Load the value from TDR into the TCNT register. -Set the conditions for operation, including disabling the LOAD bit, setting the count-up bit, configuring the internal clock to be equivalent to pclk × 2, and finally enabling the EN bit to put Timer into operation. -Thread 1 and Thread 2 run count-up in parallel, but Thread 2's duration is 2/3 of Thread 1's. -At the end of tasks, each thread checks the overflow status and displays the appropriate message indicating a fault or normal operation.*				
7	countup_forkjoin_pclk4.v		*The job task is the same as the task in Test 6. However, the count-up with the internal clock equivalent to pclk × 4.*	*The job description is the same as the task description in Test 6. However, there is a slight difference: setting the count-up with the internal clock equivalent to pclk × 4.*				
8	countup_forkjoin_pclk8.v		*The job task is the same as the task in Test 6. However, the count-up with the internal clock equivalent to pclk × 8.*	*The job description is the same as the task description in Test 6. However, there is a slight difference: setting the count-up with the internal clock equivalent to pclk × 8.*				
9	countup_forkjoin_pclk16.v		*The job task is the same as the task in Test 6. However, the count-up with the internal clock equivalent to pclk × 16.*	*The job description is the same as the task description in Test 6. However, there is a slight difference: setting the count-up with the internal clock equivalent to pclk × 16.*				
10	countdw_forkjoin_pclk2.v	**Count Down**	*-Read a random number less than 255. -Thread 1: Counts down with an internal clock of pclk × 2, starting from the number above. If the Thread 1 count-down goes below 0, it checks if the underflow status is triggered. If so, it displays a "pass" message (normal operation); otherwise, it displays a "faulty" message. -Thread 2 runs counting-down in parallel with Thread 1, but Thread 2's job duration is shorter than Thread 1's (about 2/3 of Thread 1's duration). -Thread 2 eventually stops its counting job (while Thread 1 is still running) and checks if an underflow occurred. If underflow occurs, it displays a "faulty" message; otherwise, it displays a "pass" (normal operation).*	*-Write a random value to TDR (address = 0). -Load the value from TDR into the TCNT register. -Set the conditions for operation, including disabling the LOAD bit, setting the count-down bit, configuring the internal clock to be equivalent to pclk × 2, and finally enabling the EN bit to put Timer into operation. -Thread 1 and Thread 2 run count-down in parallel, but Thread 2's duration is 2/3 of Thread 1's. -At the end of tasks, each thread checks the underflow status and displays the appropriate message indicating a fault or normal operation.*				
11	countdw_forkjoin_pclk4.v		*The job task is the same as the task in Test 10. However, the count-down with the internal clock equivalent to pclk × 4.*	*The job description is the same as the task description in Test 10. However, there is a slight difference: setting the count-down with the internal clock equivalent to pclk × 4.*				

12	countdw_forkjoin_pclk8.v		*The job task is the same as the task in Test 10. However, the count-down with the internal clock equivalent to pclk × 8.*	*The job description is the same as the task description in Test 10. However, there is a slight difference: setting the count-down with the internal clock equivalent to pclk × 8.*				
13	countdw_forkjoin_pclk16.v		*The job task is the same as the task in Test 10. However, the count-down with the internal clock equivalent to pclk × 16.*	*The job description is the same as the task description in Test 10. However, there is a slight difference: setting the count-down with the internal clock equivalent to pclk × 16.*				
14	countup_pause_countup_pclk2.v	**Pause**	*-Enter a random number less than 255.* *-The timer is enabled and counts up with an internal clock of pclk × 2, starting from the number read above.* *-It runs for a while, then pauses for a duration of COUNT_PAUSE time units.* *-After the pause, it checks if an overflow occurred. If an overflow is detected, it displays a "faulty" message; otherwise, it displays a "pass" (normal operation).* *-Keep the operation condition unchanged for the timer to count up and set the EN bit to 1. The timer continues counting up and, upon reaching 255, transfes to 0.* *-Check if the overflow status is triggered. If so, display a "pass" message (normal operation); otherwise, display a "faulty" message.*	*-Write a random value to the TDR (address = 0).* *-Load the value from TDR into the TCNT register.* *-Set the conditions for operation, including disabling the LOAD bit, setting the count-up bit, configuring the internal clock to be equivalent to pclk × 2, and finally enabling the EN bit to put the timer into operation.* *-The timer runs and counts up for a while, then disable the EN bit to stop the timer during the PAUSE time units.* *-After the pause, check the overflow status and display the appropriate message: indicate a fault if overflow is triggered, or normal operation if overflow has not yet been triggered.* *-Enable the timer (set the EN bit to 1). The timer continues counting up from where it stopped. Once the timer exceeds 255, check if the overflow status is triggered. If so, display a "pass" message (normal operation); otherwise, display a "faulty" message.*				
15	countdw_pause_countdw_pclk2.v		*Enter a random number less than 255.* *-The timer is enabled and counts down with an internal clock of pclk × 2, starting from the number read above.* *-It runs for a while, then pauses for a duration of COUNT_PAUSE time units.* *- After the pause, it checks if an underflow occurred. If an underflow is detected, it displays a "faulty" message; otherwise, it displays a "pass" (normal operation).* *- Keep the operation condition for the timer as before: counting-down, pclk2 and set the EN bit to 1.* *-The timer continues counting down and, upon reaching below 0, converts to 255.* *-Check if the underflow status is triggered. If so, display a "pass" message (normal operation); otherwise, display a "faulty" message.*	*-Write a random value to the TDR (address = 0).* *-Load the value from TDR into the TCNT register.* *-Set the conditions for operation, including disabling the LOAD bit, setting the count-down bit, configuring the internal clock to be equivalent to pclk × 2, and finally enabling the EN bit to put the timer into operation.* *-The timer runs and counts down for a while, then disable the EN bit to stop the timer during the PAUSE time units.* *-After the pause, check the underflow status and display the appropriate message: indicate a fault if underflow is triggered, or normal operation if underflow has not yet been triggered.* *-Enable the timer (set the EN bit to 1). The timer continues counting down from where it stopped. Once the timer goes below 0, transfers to 255, check if the underflow status is triggered. If so, display a "pass" message (normal operation); otherwise, display a "faulty" message.*				
16	countup_reset_countdw_pclk2.v	**Reset**	*-Read a random number less than 255.* *-The timer is enabled and counts up with an internal clock of pclk × 2, starting from the number read above.* *-It runs for a while, then trigger reset signal in order that TDR,TCR, TSR are set to their default value (0).* *- After the reset, it checks if registers TDR,TCR,TSR equals default values (0). If they equal 0, it displays a "pass" message (normal operation), otherwise, it displays a "failed"* *- Set the timer to count down, write a value similar to the one before the reset, and set the EN bit to 1.* *-The timer begins counting down and, upon reaching below 0, changes to 255.* *-Check if the underflow status is triggered. If so, display a "pass" message (normal operation); otherwise, display a "faulty" message.*	*-Write a random value to the TDR (address = 0).* *-Load the value from TDR into the TCNT register.* *-Set the conditions for operation, including disabling the LOAD bit, setting the count-up bit, configuring the internal clock to be equivalent to pclk × 2, and finally enabling the EN bit to put the timer into operation.* *-The timer runs and counts up for a while,then trigger reset signal in order that TDR,TCR, TSR are set to their default value (0).* *-After the reset, it checks if registers TDR,TCR,TSR equals default values (0). If they equal 0, it displays a "pass" message (normal operation), otherwise, it displays a "failed" .* *-Set the operation condition for the timer to count down (set TCR[5]), put value similar to the one before the reset, into TCNT and enable the timer (set the EN bit to 1).* *-The timer begins counting down. Once the timer goes 0, transfers to 255, check if the underflow status is triggered. If so, display a "pass" message (normal operation); otherwise, display a "faulty" message.*				
17	countdw_reset_countup_pclk2.v		*-Read a random number less than 255.* *-The timer is enabled and counts down with an internal clock of pclk × 2, starting from the number read above.* *-It runs for a while, then trigger reset signal in order that TDR,TCR, TSR are set default value (0).* *- After the reset, it checks if registers TDR,TCR,TSR equals default values (0). If they equal 0, it displays a "pass" message (normal operation), otherwise, it displays a "failed"* *-Set the operation condition for the timer to count up, write a number to be continued counting and set the EN bit to 1.* *-The timer begins counting up and, upon reaching 255, over to 0.* *Check if the overflow status is triggered. If so, display a "pass" message (normal operation); otherwise, display a "faulty" message.*	*-Write a random value to the TDR (address = 0).* *-Load the value from TDR into the TCNT register.* *-Set the conditions for operation, including disabling the LOAD bit, setting the count-down bit, configuring the internal clock to be equivalent to pclk × 2, and finally enabling the EN bit to put the timer into operation.* *-The timer runs and counts down for a while, then disable the EN bit to stop the timer during the PAUSE time units.* *-After the pause, it checks if registers TDR,TCR,TSR equals default values (0). If they equal 0, it displays a "pass" message (normal operation), otherwise, it displays a "failed" .* *-Set the operation condition for the timer to count down (set TCR[5]), enable the timer (set the EN bit to 1).* *-The timer begins counting down from where it stopped. Once the timer goes 0, transfers to 0, check if the underflow status is triggered. If so, display a "pass" message (normal operation); otherwise, display a "faulty" message.*				
18	countup_reset_load_countdw_pclk2.v	**Load**	*The job task is the same as in Test 16. However, after the reset, load a new random number into the timer and start counting down with an internal clock equivalent to PCLK × 2.*	*The job description is the same as the task description in Test 16. However, there is a slight difference: after the reset, set the timer with a new random number, and continue counting down with an internal clock equivalent to PCLK × 2.*				
19	countdw_reset_load_countdw_pclk2.v		*The job task is the same as in Test 16. However, after the reset, load a new random number into the timer and start counting down with an internal clock equivalent to PCLK × 2.*	*The job description is the same as the task description in Test 16. However, there is a slight difference: after the reset, set the timer with a new random number, and continue counting down with an internal clock equivalent to PCLK × 2.*				
20	fake_underflow.v	**Fake**	*-Timer is disabled, set Timer with pclk2, couting-down.* *-Load 0 into TCNT.* *-Load 255 into TCNT.* *-Check if Underflow (TSR[1]) is triggered or not. And display appropriate message*	*-Write 8'h00 to TDR. Load data from TDR into TCNT* *-make Timer disabled; off load function; setting internal_clock equivalent to pclk2; set count-down.* *-Write 8h'FF (255) to TDR. Load data from TDR into TCNT* *-Timer still disabled; Off load function* *-Check TSR[1] if underflow or not.*				
21	fake_overflow.v		*-Timer is disabled, set Timer with pclk2, couting-up.* *-Load 255 into TCNT.* *-Load 255 into TCNT.* *-Check if Overflow (TSR[0]) is triggered or not. And display appropriate message*	*-Write 8'hFF to TDR. Load data from TDR into TCNT* *-make Timer disabled; off load function; setting internal_clock equivalent to pclk2; set count-up.* *-Write 8h'00 to TDR. Load data from TDR into TCNT* *-Timer still disabled; Off load function* *-Check TSR[1] if underflow or not.*				