

LAB - 8

Date: 1st April, 2020

Lab Title: Ring Counters

Recorded and Submitted by: Bheshaniya Abhishek

Objective: To write VHDL program for ring counters, Overbars counter and Twisted bars counters.

Implementation

We will use two switch buttons to make counters one for count up and one for reset the counter.

We will use 4 Led for ring counter and 4 Led for J-K counter to show output.

Hex Seven segment display will display the ^{States} counts. Hex0 for ring counter and Hex2 for J-K.

Procedure:

- 1st step is to make table for output for corresponding input.
- then define inputs and outputs in entity section
- we will use signal for both input and output.

Signal Ring: std_logic_vector(3 downto 0);

signal J_K: std_logic_vector(3 downto 0);

- where ever there is a reset occurs load both the counters with default values.
- we will use case statement to display the state on seven segment LED.

- After writing entire code we will compile the code and will do pin assignment.

- After doing pin assignment we will begin this Project.

- Then we will load the program in to fpga board and will test for different case.

Table for Straight ring / overbeck Counter

State	Q ₀	Q ₁	Q ₂	Q ₃
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
0	1	0	0	0

Table for Twisted ring / Johnson Counter

State	Q ₀	Q ₁	Q ₂	Q ₃
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	0	1	1	1
5	0	0	1	1
6	0	0	0	1
7	0	0	0	0
0	0	0	0	0

Pin Assignment

60

Input / Output Variables	Signal Name	Pin no.
EN	SW [9]	PIN - AE12
CLEAR	KEY [0]	PIN - AA14
CLOCK	KEY [1]	PIN - AA15
J [3]	LEDR [9]	PIN - Y21
J [2]	LEDR [8]	PIN - W21
J [1]	LEDR [7]	PIN - W20
J [0]	LEDR [6]	PIN - Y19
R [3]	LEDR [3]	PIN - V18
R [2]	LEDR [2]	PIN - V17
R [1]	LEDR [1]	PIN - W16
R [0]	LEDR [0]	PIN - V16
HEX0 [6]	HEX0 [0]	PIN - AE26
HEX0 [5]	HEX0 [1]	PIN - AE27
HEX0 [4]	HEX0 [2]	PIN - AE28
HEX0 [3]	HEX0 [3]	PIN - AG27
HEX0 [2]	HEX0 [4]	PIN - AF28
HEX0 [1]	HEX0 [5]	PIN - AG28
HEX0 [0]	HEX0 [6]	PIN - AH28
HEX1 [6]	HEX2 [0]	PIN - AB29
HEX1 [5]	HEX2 [1]	PIN - AE29
HEX1 [4]	HEX2 [2]	PIN - AD29
HEX1 [3]	HEX2 [3]	PIN - AC28
HEX1 [2]	HEX2 [4]	PIN - AP30
HEX1 [1]	HEX2 [5]	PIN - AC29
HEX1 [0]	HEX2 [6]	PIN - AC30

VHDL Program:-

```
library ieee;
use ieee.std-logic-1164.all;
use ieee.numeric_std.all;

entity abhesuniversity_lab8 is
  port (
    EN : in std-logic;
    CLEAR : in std-logic;
    clock : in std-logic;
    R : out std-logic-Vector (3 downto 0);
    J : out std-logic-Vector (3 downto 0);
    HEX0 : out std-logic-Vector (6 downto 0);
    HEX1 : out std-logic-Vector (6 downto 0);
  );
end abhesuniversity_lab8;

architecture COUNTER of abhesuniversity_lab8 is
  signal RING : std-logic-Vector (3 downto 0);
  signal JH0V : std-logic-Vector (3 downto 0);
begin
  process (clock, EN, CLEAR)
  begin
    if (EN = '1')
    then
      if CLEAR = '0' then
        RING <= "0001";
        JH0V <= "0000";
      end if;
    end if;
  end process;
end architecture;
```


else if Rising - edge (clock) then
 RING(1) <= RING(0);
 RING(2) <= RING(1);
 RING(3) <= RING(2);
 RING(0) <= RING(3);
 JHON(1) <= ~~JHON~~ JHON(0);
 JHON(2) <= ~~JHON~~ JHON(1);
 JHON(3) <= ~~R~~ JHON(2);
 JHON(0) <= NOT JHON(3);

R <= RING;
 J <= JHON;

Case RING is

when "0001" => HEX0 <= "0000001";
 when "0010" => HEX0 <= "1001111";
 when "0100" => HEX0 <= "0010010";
 when "1000" => HEX0 <= "0000110";
 when others => HEX0 <= "1111111";

end case;

Case JHON is

when "0000" => HEX1 <= "0000001";
 when "0001" => HEX1 <= "1001111";
 when "0011" => HEX1 <= "0010010";
 when "0111" => HEX1 <= "0000110";
 when "1111" => HEX1 <= "1001100";
 when "0110" => HEX1 <= "0100100";
 when "1100" => HEX1 <= "0100000";
 when "1000" => HEX1 <= "0001111";
 when Others => HEX1 <= "1111111";

End case;

end if;

else

Ring $\leq "0001"$

THOIV <= "0000";

```
Hexo <= "111111";
```

Hex 7 <= "111111";

end if ~~if visited~~ \rightarrow ~~if not visited~~

end process; $\Pi \rightarrow \text{CPU}$ MONT

End COUNTER; DOWN \Rightarrow (0) VOLT

Discussion:

'1000000' = 2000000 - "1000" = 1999000

"111001" \Rightarrow 987H (= "0110" reversed)

$$010010010011 \Rightarrow 0x111 (= 7) \quad 00101011 \quad [50/100]$$

01100000" 23 0X3H (= "0001" 00000001)

$\Gamma \vdash A$ \Rightarrow $\exists x \in \text{dom}(\Gamma) . A(x)$

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840. 84

60

[illegible]

12 NOV 12

000005 11 → CY34 / 1000 12 13 14 15 16 17 18 19

11/11/11

10/100 1" 50 1000 1000000

Wahrentz 0017-79 1979 1979

 ~~$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ 0 & 1 \end{pmatrix}$~~

EXHIBIT (2) "1111" CONVO

$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ 0 & 1 \end{pmatrix}$ is "unitary" complex

$$0 \leq 0.11 \Rightarrow E(x) = 0.11 \text{ (from (c))}$$

$\frac{000}{1} = \frac{000}{1} \times \frac{10^6}{10^6} = \frac{000}{1} \times 10^6$

$\text{HCl} + \text{NaOH} = \text{NaCl} + \text{H}_2\text{O}$

$\frac{1}{2} \times 20 = 10$

[illegible]

1711-1957