

Конвертор бинарног броја у број у природном BCD коду

Алгоритам конверзије се састоји од следећих корака:

- 1) Ако је било која цифра (стотине, десетице, јединице) већа од 4 додати 3 на ту цифру
- 2) Померити све бите за једно место улево
- 3) Ако је већ обављено 8 померања процес конверзије је завршен
- 4) Поновити корак 1.

Пример конверзије броја 178 (10110010) из бинарног у BCD коришћењем горе описаног алгоритма је приказан на слици 1.

<i>shift_cnt</i>	<i>shift_reg</i>				
7	0000	0000	0000	10110010	
6	0000	0000	0001	01100100	
5	0000	0000	0010	11001000	
4	0000	0000	0101	10010000	
4	0000	0000	1000	10010000	корекција!
3	0000	0001	0001	00100000	
2	0000	0010	0010	01000000	
1	0000	0100	0100	10000000	
0	0000	1000	1001	00000000	
0	0000	1011	1100	00000000	корекција!
0	0001	0111	1000	00000000	
	1	7	8		

Слика . Пример конверзије из бинарног у BCD формат

Описани алгоритам се може једноставно имплементирати коришћењем коначне машине стања. Из алгоритма се јасно издвајају 3 стања: почетно стање (*Idle*), стање померања (*Shift*) и стање корекције, односно сабирања (*Add*).

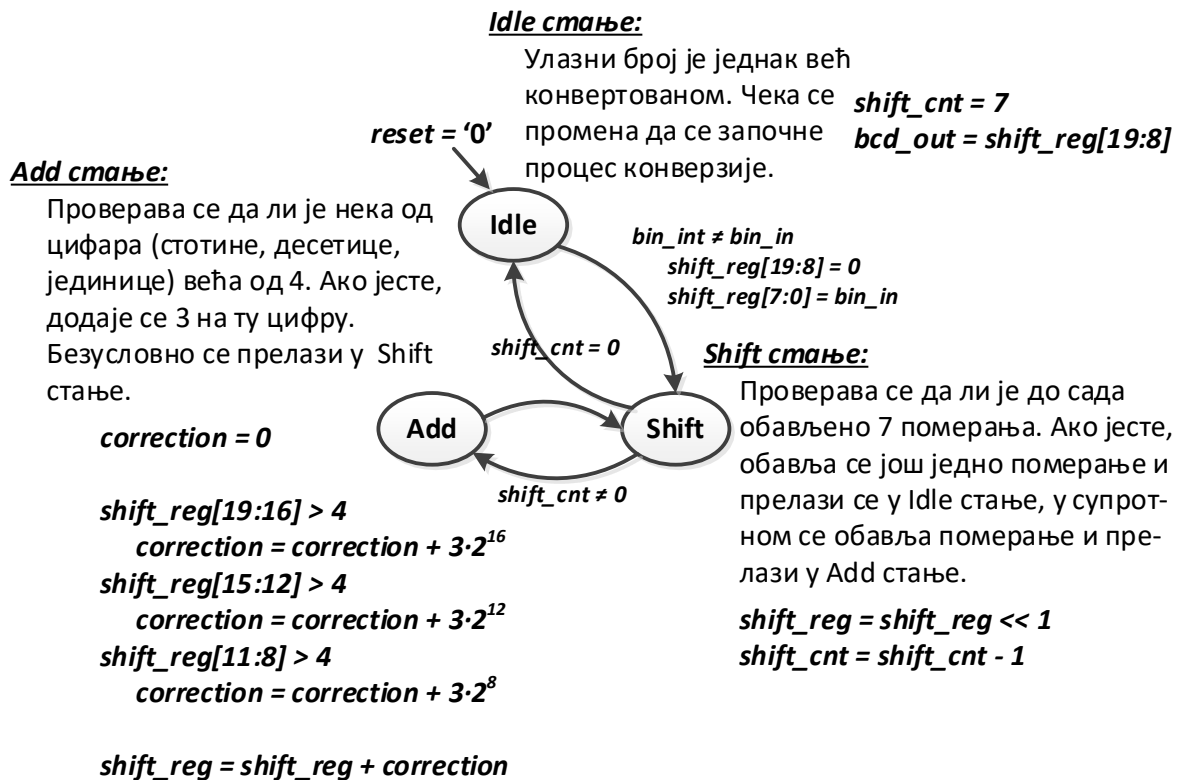
Машина стања се налази у почетном стању док год је улазни бинарни број једнак броју уписаном у интерни регистар који представља бинарну вредност тренутног BCD излаза. У тренутку када се детектује да се бинарни број који се налази у интерном регистру и улаз разликују значи да излаз више није валидан па је потребно започети нови процес конверзије иницијализовањем свих интерних сигнала и преласком у стање померања.

У стању померања је потребно проверити да ли је до сада обављено 7 померања. Ако јесте обавља се још 1 померање чиме је процес конверзије завршен и прелази се у почетно стање. У супротном се обавља померање, бројач преосталих померања се умањује за 1 и прелази се у стање корекције.

У стању корекције проверава се вредност сваке од цифара и додаје одговарајући фактор на корекциони члан. Корекциони члан је потребно дефинисати унутар самог стања корекције пошто он представља променљиву локалног карактера. Обратите пажњу на коришћење променљиве у коду из прилога упутства.

Архитектура описане машине стања приказана је на слици 2. Оператори и типови података око самог дијаграма стања не одговарају VHDL-у, већ само представљају додатни опис функционалности.

Обратити пажњу у коду да се излазни BCD број мења у секвенцијалном процесу јер је потребно да његова вредност остане непромењена у току конверзије, а мења се тек на крају конверзије.



Слика 2. Машина стања којом се реализује конверзија 8-битног бинарног броја у BCD формат