

# ТЕСТВАНЕ И ДИАГНОСТИКА

## Упражнение 8

### Методи, базирани върху симулации - CONTEST алгоритъм

/Concurrent Test Generator for Sequential Testing/

Алгоритъма включва 3 фази:

1. **Инициализация.** Инициализират се тригерите на последователната схема.
2. **Откриване на конкурентни повреди.** Създава се списък с повреди. На базата на избран входен вектор се симулира поведението на схемат без и със повреди. Всички повреди, които се откриват в избрания вектор се изключват от списъка на повредите. Избира се следващ вектор до достигане на определено покритие на повредите.
3. **Откриване на единични повреди.** Търсят се вектори за всяка една повреда по отделно.

#### Фаза 1. Инициализация

Работи се с изправна схема. Целта е да се инициализират всички тригери с минимални усилия. При много сложни вериги се допуска до 10% от тригерите да останат неинициализирани. Стартира се от начален вектор, избран от тестващия или избран на случаен принцип. Блоквата схема на алгоритъма е показана на фиг.1.

#### Фаза 2. Откриване на конкурентни повреди

Създава се списък от повреди от тип слепване (всички или избрано подмножество). За всеки входен вектор се прави оценка ( $C_i$ ) за всяко слепване в списъка. Обикновено се тръгва от инициализационния вектор, защото той може да е активирал или дори открил повреди. Симулира се повредена схема и се проследява разпространението на ефекта от повредата. Оценява се разстоянието от последната връзка, в която се е проявила повредата до най-близкия първичен изход, като за всеки комбинационен елемент се добавя 1, а за всеки тригер – 100. Резултатът се записва в  $C_i$ .

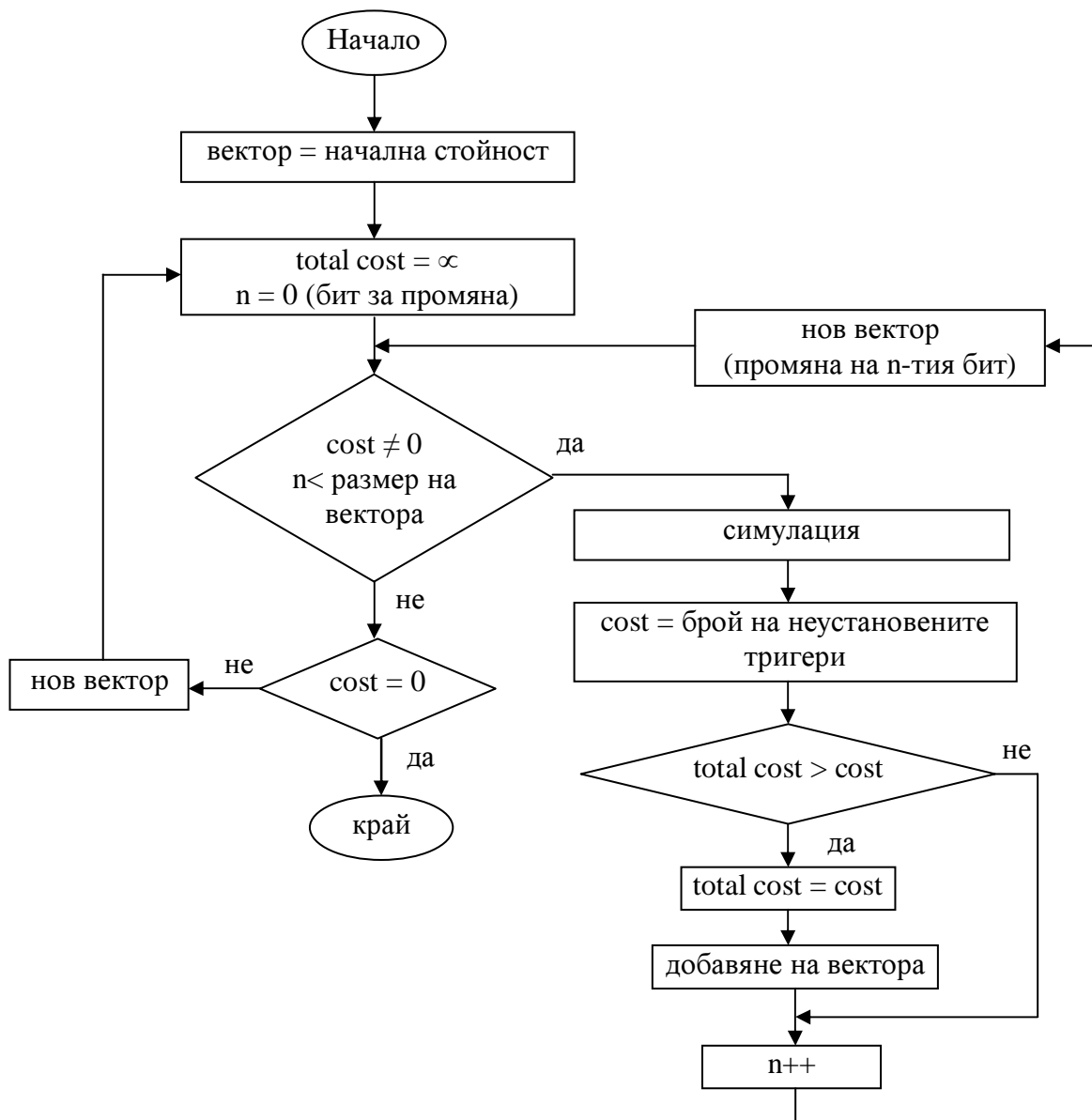
Всички повреди, за които  $C_i = 0$  се маркират като „открити” и се изключват от списъка с повредите.

Стойностите  $C_i$  за всички останали повреди се запомнят, тестващият вектор се променя и се изчисляват новите стойности за  $C_i'$ . Ако  $C_i' < C_i$ , векторът се счита за приемлив и се добавя към тестващата последователност.

От гледна точка ускоряване на алгоритъма не се анализира  $C_i' < C_i$  за всяко  $i$ , а се вземат най-ниските 10% от оценките  $C_i$ . Ако това условие не е изпълнено, векторът се отхвърля и се избира следващия вектор.

Когато списъкът с повредите остане празен, процедурата се прекратява и последователността от приемливи вектори представлява тестващата последователност за тази схема. Когато останат единични повреди, за които не може да се постигне  $C_i' < C_i$ , процедурата се прекратява и се преминава към фаза 3.

Алгоритъмът се ускорява допълнително като за всеки елемент предварително се прави оценка на разстоянието до най-близкия първичен изход.



фиг.1. Инициализация

### Фаза 3. Откриване на единични повреди.

Използва се подхода SCOAP - за всеки сигнал се изчисляват динамична контролируемост по отношение на 0 и 1 и динамична наблюдаемост. Изчисляват се:

- комбинационна динамична контролируемост- отразява колко първични входа трябва да променят състоянието си, за да може дадена връзка да постигне желаното състояние;
  - комбинационна динамична контролируемост на  $i$ -тата връзка по отношение на 0:  $DCC0(i)$  ;
  - комбинационна динамична контролируемост на  $i$ -тата връзка по отношение на 1:  $DCC1(i)$  ;
- Последователна динамична контролируемост- отразява броя необходими вектори, за да може дадена връзка да постигне желаното състояние;

- последователна динамична контролируемост на  $i$ -тата връзка по отношение на 0:  $DSC0(i)$  ;
- комбинационна динамична контролируемост на  $i$ -тата връзка по отношение на 1:  $DSC1(i)$  ;