БИЛЕТ N 1

1. Понятия «контроль» и «диагностика» в теории диагностирования СВТ. Методы контроля и диагностики.

Под контролем СВТ принято понимать процессы, обеспечивающие обнаружение ошибок в работе, вызванных отказом или сбоем.Под диагностикой подразумевается процедура локализации неисправности объекта, т.е. установления того, какая часть объекта контроля (ОК) является неисправной. При диагностировании производится локализация неисправности на более низком иерархическом уровне, чем при контроле. В некоторых случаях под процедурой диагностики подразумевается анализ результатов измерений некоторой совокупности параметров, на основе которого могут приниматься решения о техническом состоянии объекта, области его функциональной применимости, режимах эксплуатации. Из многообразия направлений следует особо выделить тестовый контроль (ТК), функциональный контроль (ФК) и параметрический контроль (ПК). Процедура тестового контроля осуществляется в специально отведенные для нее промежутки времени, в течение которых ОК не эксплуатируется по назначению. Достоинство ТК состоит в его универсальности, т.е. возможности с помощью аналогичных по сути методов реализовать контроль и диагностику на всех стадиях жизненного цикла изделия. Функциональный контроль, который часто называют аппаратным или схемным, не предусматривает генерации специальных тестовых воздействий. Этот вид контроля используется на стадии эксплуатации. Параметрический контроль предполагает оценку состояния ОК по косвенным признакам, которые имеют интегральный характер и чаще всего выражаются непрерывными величинами..

2. Аддитивный критерий оптимизации.

Целевая функция – путем сложения нормированных значений частных критериев. Нормированные значения представляют собой отношение реального значения нормального критерия к некоторой нормирующей величине, измеряющейся в тех же единицах, что приводит к безразмерной величине данного критерия.

1) В качестве делителей выбираем директивные значения параметров, заданное заказчиком. Недостаток: заданная в ТЗ величина рассматривается как образцовая, что не всегда полезно.

2) Выбираем экстремальные значения критериев, достигаемых в области существующих проектных решений.

3) Выбираем разность между минимальным и максимальным параметром в области существующих проектных решений.

Выбор варианта является субъективным решением

$$F(x)=\sum_{i=1}^{n}\frac{F_i(x)}{F_i^H(x)}=\sum_{i=1}^{n}f_i(x)$$

Необходимо учитывать важность параметров, добавим коэффициент $F(x)=\sum_{i=1}^{n}c_i f_i(x)$

$F_i^H(x)$ – нормирующий делитель от частного коэффициента.

Недостаток такого критерия:

– нет логической взаимосвязи между оцениваемыми параметрами

– при вычислении данного критерия может происходить взаимная компенсация значений параметров

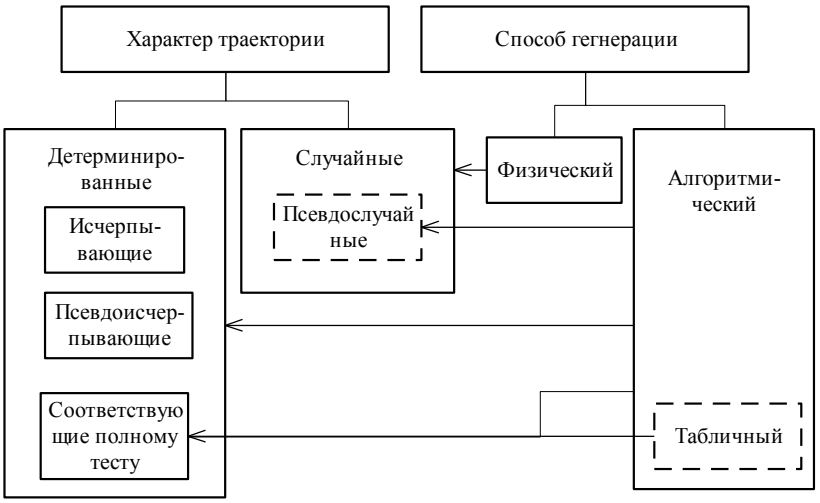
Ограничения на экстремальные значения параметров и коэффициентов значимости – является машинным приемом для точного определения критерия.

БИЛЕТ N 2

1. Общая характеристика тестового контроля СВТ.

Организация тестового диагностирования связана с наличием специально подготовленных воздействий и эталонных реакций на эти воздействия, позволяющих определить техническое состояние ОД и, в случае его неисправности, локализовать место неисправности, а также специальной аппаратуры, предназначенной для генерации входных воздействий и регистрации выходных реакций ОД с их последующим анализом.

На рис.3.1 представлена классификация испытательных последовательностей. По характеру траектории они могут быть разделены на детерминированные и случайные. Детерминированные последовательности характеризуются тем, что к моменту начала тестового эксперимента их траектория однозначно определена. К ним можно отнести исчерпывающие, псевдоисчерпывающие и соответствующие понятию полного теста последовательности. Исчерпывающая последовательность представляет собой полный перебор всех возможных входных воздействий. Следовательно, с ее помощью реализуется тест $T \subseteq C^U \times C^U$, который априорно обладает полнотой.



В большинстве случаев ОК можно разделить на ряд независимых частей, каждую из которых можно рассматривать как отдельный объект контроля. Если число входов для каждого из них оказывается меньше, чем для всего устройства в целом, то длина исчерпывающей последовательности может быть сокращена. Такая последовательность называется псевдоисчерпывающей и , по сути, является исчерпывающей для каждой из частей ОК, тестирование которых может производиться независимо друг от друга.

2. Мультипликативный критерий оптимизации.

Используется принцип справедливой компенсации абсолютных значений нормированных частных критериев. В целом ряде задач используется не абсолютное значение, а принцип относительной компенсации, то есть справедливым следует считать такой компонент, когда суммарный уровень относительного снижения значений одного или нескольких критериев не превышает суммарного уровня относительного увеличения значений других критериев.

$\sum_{i=1}^{n} \frac{\Delta F_i(x)}{F_i(x)} = 0$, где n – количество параметров, $F_i(x)$ – начальное значение частного коэффициента,

$\Delta F_i(x)$ –изменение значений в новом варианте.

С учетом коэффициента значимости $\sum_{i=1}^{n} C_i \frac{\Delta F_i(x)}{F_i(x)} = 0$

Формулу можно преобразовать

$$\sum_{i=1}^{n} \frac{\Delta F_i(x)}{F_i(x)} = \sum_{i=1}^{n} d\left(\ln F_i(x)\right) = d\left(\ln \prod_{i=1}^{n} F_i(x)\right) = 0$$

Это всё сложно реализовать, поэтому необходимо варьировать параметрами в произведении.

$F_i(x) = \prod_{i=1}^{n} F_i(x)$ – критерий является мультипликативным.

С учетом коэффициента значимости используют $F_i(x) = \prod_{i=1}^{n} C_i F_i(x)$

Недостаток такого критерия:

– нет логической взаимосвязи между оцениваемыми параметрами

– при вычислении данного критерия может происходить взаимная компенсация значений параметров

БИЛЕТ N 3

1. Требования, предъявляемые к тестам.

Достаточная полнота контроля. Полный тест – тест, реализация которого позволяет обнаружить все возможные неисправности из заданного класса неисправностей Количественная оценка полноты контроля есть отношение количества выявленных неисправностей к общему числу возможных неисправностей заданного класса.

Приемлемое время испытаний объекта контроля. Время контроля непосредственно связано с количеством элементарных проверок или *емкостной* сложностью теста прямой (не обязательно линейной) пропорцией. Поэтому наиболее выгодно применение *минимальных* тестов – полных тестов, имеющих минимально возможное количество элементарных проверок, или, по крайней мере, не избыточных тестов – полных тестов, исключение из которых любой элементарной проверки нарушает полноту теста.

Малая трудоемкость генерации тестов. Создание тестов, эффективных по показателю полноты и количеству элементарных проверок, представляет собой сложную задачу. Ее решение для большинства аппаратных компонент СВТ возможно только с помощью автоматизированных систем генерации тестов (АСГТ).

Глубина диагностирования. Очевидным требованием к диагностическим тестам является локализация места неисправности с точностью до конкретного сменного элемента или до определения области применимости с ограничением функций для «неразборных» объектов, например БИС. При этом необходимо отметить, что в процессе проектирования систем технического диагностирования (СТД) необходимо, чтобы:

- алгоритмы технического диагностирования обеспечивали в первую очередь требуемую достоверность результатов определения технического состояния объектов диагностирования при заданных ограничениях на другие параметры СТД;
- проектирование СТД осуществлялось одновременно с проектированием объекта диагностирования;
- при проектировании объектов диагностирования предусматривалось обеспечение контролепригодности и достаточной глубины их диагностирования;

способ генерации входных последовательностей (см ниже) и метод диагностирования соответствовали аппаратуре контроля.

2. Минимаксные критерии оптимизации.

Критерий основан на принципе компромиссности, используется идея равномерности.

Необходимо найти такой вариант, у которого все относительные значения всех критериев будут одинаковые. $f_i(x) = k \;\; i = \overline{1,n} \;\; x = \{x_1 \dots x_n\} \;\; k = const$ или с учетом значимости $c_i f_i(x) = k$

Необходимо найти такую совокупность значений параметров, чтобы целевая функция принимала наихудшие значения параметров, чтобы в общем функция принимала максимальные значения.

$$F(x) = \max_{F(x)} \min_i f_i(x) \;\; i = \overline{1,n} \;\; x = \{x_1 \dots x_n\}$$

Ещё называется принципом гарантированного результата

$F(x) = \min_{F(x)} \max_i f_i(x)$ – минимаксный выбор

При учете значимости $F(x) = \min_{F(x)} \max_i \left(f_i(x) c_i \right)$

Наиболее часто используемые методы нахождения экспертных оценок: ●метод ранжирования; ●метод присвоения баллов.

Метод ранжирования: собирается l экспертов, им предлагается расставить n критериев по рангу, причем самый важный принимает n-ый ранг, а наименее важный – 1-ый ранг.

Ранг каждого элемента определяется

$C_i = \dfrac{\sum_{k=1}^{l} r_i^k}{\sum_{k=1}^{l} \sum_{i=1}^{n} r_i}$, где r – ранг, а C_i – значимость параметра, r_i^k – ранг i–го критерия выставленный k–ым экспертом.

Метод приписывания баллов. Также эксперты проставляют баллы от 0 до 10. Несколько параметров могут иметь одинаковые баллы и могут использовать дробные числа.
H_i^k – балл i–го элемента выставленный k–ым экспертом.

$H_i^k = \frac{h_i^k}{\sum_{i=1}^{n} h_i^k}$, $\sum_{i=1}^{n} h_i^k$ – сумма балов выставленных k–ым экспертом всем элементам

$$C_i = \frac{\sum_{k=1}^{l} H_i^k}{\sum_{k=1}^{l}\sum_{i=1}^{n} H_i^k}$$

Для более точной характеристики необходимо учитывать значимость (компетентность) эксперта.
μ – коэффициент компетентности

$$C_i = \frac{\sum_{k=1}^{l} \mu_k H_i^k}{\sum_{k=1}^{l}\sum_{i=1}^{n} \mu_k H_i^k}$$

БИЛЕТ N 4

1. Краткая характеристика процесса проектирования.

Проектирование – это разработка технической документации изделия, согласно функциональным требованиям или создание, преобразование и представление принятой формы образа еще не существующего объекта.

Требования – имеются ли физические средства для реализации, может уже кто-то придумал (технология), техническое задание. Проектирование включает комплекс работ исследовательского, расчетного и конструкторского характера. В ТЗ описываются функции, которые должен выполнять объект. После этого описывается назначение, область применения и т.д.

Способы: ручной, автоматизированный, автоматический. Автоматизированный – компьютер выполняет большинство задач, больше или меньше в зависимости от стадии. Автоматических систем на данный момент не существует.

2. Алгоритм простой итерации (2-ая модель сигналов).

Существуют две реализации алгоритма простой итерации: последовательная реализация, параллельная реализация. При последовательной реализации алгоритма простой итерации необходимо задать начальное состояние схемы, затем подать первый входной набор и вычислить значения всех выходов логических элементов схемы на данном наборе. Если результат текущей итерации не равен результату предыдущей итерации, выполняется следующая итерация. Процесс заканчивается при полном совпадении результатов двух соседних итераций либо при достижении заданного максимального числа итераций. Максимальное число итераций в данном случае равно длине максимальной цепочки логических элементов (включая ОС) плюс 1. Также признаком конца процесса моделирования может служить наличие в таблице моделирования записи результата текущей итерации, находящейся на месте, отличающегося от предпоследнего. При параллельной реализации одновременно моделируют несколько входных наборов. Число одновременно моделируемых наборов зависит от языка программирования и особенностей реализации алгоритма. Ускорение происходит за счет меньшего числа обращений к описанию схемы.

В данном алгоритме значение на любом элементе в схеме считается как функции, которая зависит от входного набора и значений элементов на предыдущей итерации:

$Y_{ij}^{k} = F\left(x^{k}, Y_{i-1}\right)$, где

$x = \{x_1, x_2, ..., x_l\}$ - входной набор (множество входных сигналов);

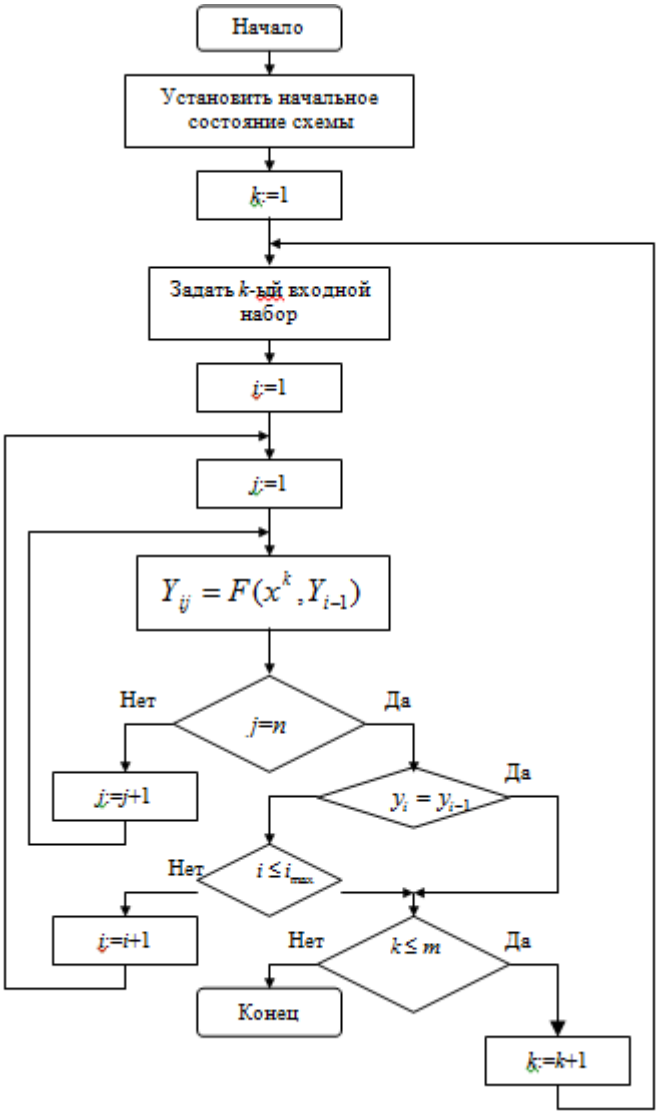
$Y = \{y_1, y_2, ..., y_n\}$ - выходные сигналы схемы;

j - номер элемента;

i - номер итерации;

k - номер входного набора ($k = \overline{1..m}$).

Схема алгоритма.



Последовательная реализация алгоритма простой итерации:

- преимущество: простота реализации;
- недостаток: большое время вычисления, большие вычислительные затраты.

Параллельная реализация алгоритма простой итерации:

- преимущество: простота реализации, ускорение за счет меньшего числа обращений к описанию схемы;
- недостаток: большое время вычисления, большие вычислительные затраты.

БИЛЕТ N 5

1. Виды обеспечений САПР.

Техническое – совокупность взаимосвязанных и взаимодействующих технических средств, необходимых для выполнения АП.

Математическое – математические модели объектов проектирования, методы и алгоритмы выполнения процедур. Бывает инвариантное (для любых) и специализированное (для конкретных задач).

Лингвистическое – совокупность всех языков, использовавшихся при проектировании. Они делятся на 2 большие группы: языки программирования (универсальные, объектно-ориентированные), и языки проектирования (на которых описаны данные).

Программное – программы и соответствующая документация для реализации САПР.

- Общесистемное (ОС) – организация функционирования технических средств, т.е. для планирования, управления и выполнения вычислительных процессов, распределения ресурсов и т.д.
- Прикладное – реализация мат. обеспечения непосредственно для выполнения проектных процедур. Обычно представлено в виде пакетов, каждый из которых обеспечивает этап проектирования или часть задач, выполняемых на некотором этапе.
- Базовое – включает модули, обеспечивающие правильное выполнение прикладного ПО.

Информационное – включает все данные, необходимые для выполнения АП. Центральным ядром является банк данных (знаний) – совокупность средств, необходимых для централизованного накопления данных. База – чисто данные. Банк = база + СУБД. Знания = база + СУБД + методы преобразования данных.

Методическое - документы, устанавливающие состав, правила отбора и эксплуатации средств АП.

Организационное – регламентирует организационную структуру проектной организации, необходимых для АП.

2. Алгоритм простой итерации (3-ая модель сигналов).

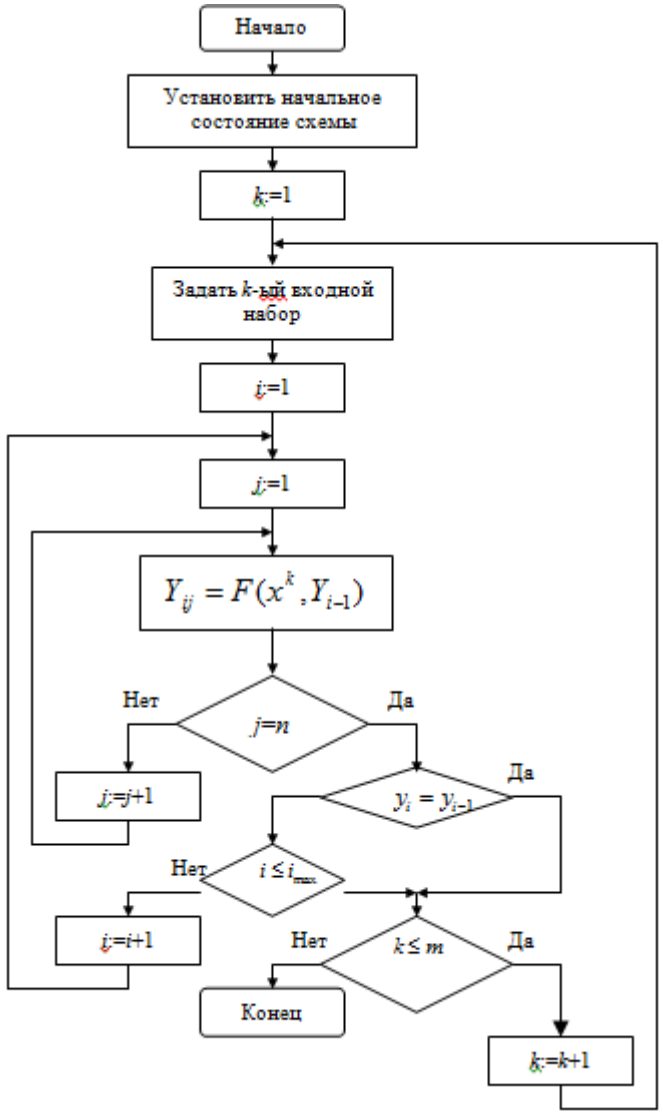
Существуют две реализации алгоритма простой итерации: последовательная реализация, параллельная реализация. При последовательной реализации алгоритма простой итерации необходимо задать начальное состояние схемы, затем подать первый входной набор и вычислить значения всех выходов логических элементов схемы на данном наборе. Если результат текущей итерации не равен результату предыдущей итерации, выполняется следующая итерация. Процесс заканчивается при полном совпадении результатов двух соседних итераций либо при достижении заданного максимального числа итераций. Максимальное число итераций в данном случае равно длине максимальной цепочки логических элементов (включая ОС) плюс 1. Также признаком конца процесса моделирования может служить наличие в таблице моделирования записи результата текущей итерации, находящейся на месте, отличающегося от предпоследнего. При параллельной реализации одновременно моделируют несколько входных наборов. Число одновременно моделируемых наборов зависит от языка программирования и особенностей реализации алгоритма. Ускорение происходит за счет меньшего числа обращений к описанию схемы.

В данном алгоритме значение на любом элементе в схеме считается как функции, которая зависит от входного набора и значений элементов на предыдущей итерации:

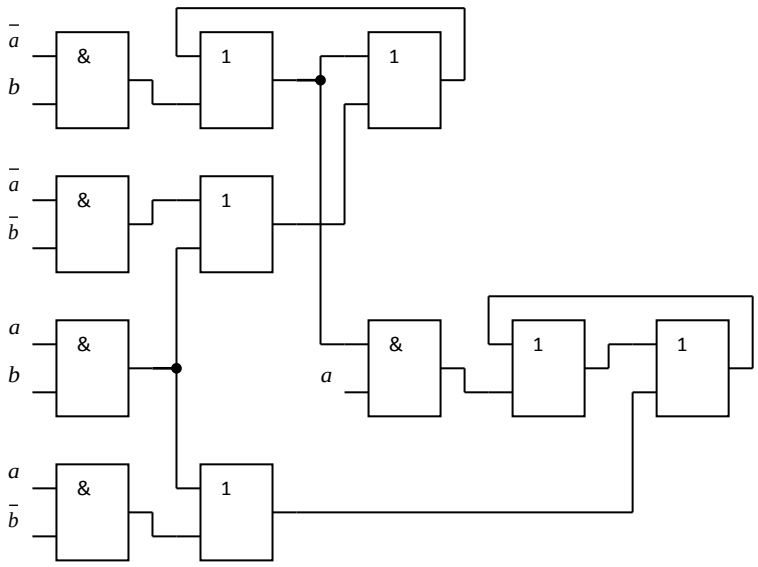
$Y_{ij}^{k} = F\left(x^{k}, Y_{i-1}\right)$, где $x = \{x_1, x_2, ..., x_l\}$ - входной набор (множество входных сигналов);

$Y = \{y_1, y_2, ..., y_n\}$ - выходные сигналы схемы; j - номер элемента; i - номер итерации; k - номер входного набора ($k = \overline{1..m}$).

Схема алгоритма.



Пример использования данного метода для троичной модели.



Максимальное число итераций $i_{\max} = 10$ (1-2-3-2-6-7-8-7-8 – самый длинный путь).

Для последовательности наборов:

00, 0x, 01, 11.

Для троичной модели начальное значение на элементах не определено.

a	*b*	1	2	3	4	5	6	7	8	9	10	11
		x	x	x	x	x	x	x	x	x	x	X
0	0	0	x	x	1	x	0	x	x	0	x	0
0	0	0	x	x	1	1	0	x	x	0	0	0
0	0	0	x	1	1	1	0	x	x	0	0	0
0	0	0	1	1	1	1	0	x	x	0	0	0
0	0	0	1	1	1	1	0	x	x	0	0	0
0	x	x	1	1	x	1	0	x	x	0	0	0
0	x	x	1	1	x	x	0	x	x	0	0	0
0	x	x	1	1	x	x	0	x	x	0	0	0
0	1	1	1	1	0	x	0	x	x	0	0	0
0	1	1	1	1	0	0	0	x	x	0	0	0
0	1	1	1	1	0	0	0	x	x	0	0	0
1	1	0	1	1	0	0	1	x	x	1	0	0
1	1	0	1	1	0	1	1	1	x	1	1	0
1	1	0	1	1	0	1	1	1	1	1	1	0
1	1	0	1	1	0	1	1	1	1	1	1	0

При использовании троичной модели в схеме можно обнаружить генераторы и состязания сигналов.
Если при походе схемы максимальное число итераций значения на некоторых элементах в схеме не установились, то данные элементы образуют генератор.
Если на одном входном наборе в схеме значение на выходе элемента принимало определенное значение, после чего было не определено, а затем приняло тоже значение, то на данном элементе идет состязание сигналов.

БИЛЕТ N 6

1. Определение САПР.

Система автоматизированного проектирования (САПР) - организационно-техническая система, состоящая из комплекса средств автоматизации проектирования (КСАП), взаимосвязанного с необходимыми подразделениями проектной организации или коллективом специалистов (пользователей системы) и выполняющая автоматизированное проектирование. Соответственно **система автоматического проектирования** выполняет автоматическое проектирование без участия человека.

САПР – совокупность средств и методов для осуществления автоматизированного проектирования, состоящее из ряда частей, наз. обеспечениями: техническое, математическое, лингвистическое, программное, информационное, методическое, организационное.

2. Алгоритм ускоренной итерации (2-ая модель сигналов).

Количество тактов (проходов) схемы (максимальное количество итераций) = количество обратных связей + 2.

При использовании данного алгоритма максимальное число итераций в последовательной схеме из 3 элементов, при изменении входного набора, будет равно 2 тактам, а не за 4 как в алгоритме простой итерации.

Для данного алгоритма:

$$Y_{ij}^{k} = F\left(x^{k}, y_{i1}, ..., y_{ij-1}, y_{i-1j}, ..., y_{i-1n}\right).$$

Если нумерация элементов в схеме неправильная нужно перенумеровать элементы.

Необходимо определить начальное состояние схемы. При вычислении значения на выходе i – того элемента на итерации h значения на входах выбираются следующим образом: если входная линия (линиями схемы будем называть внешние входы схемы и выходы элементов) имеет номер j < i, используется ее значение, полученное на итерации h, а если j > i, — значение, полученное на итерации h - 1.При использовании итераций Зейделя требуемое число итераций зависит от порядка нумерации линий схемы. Поэтому используют алгоритмы ранжирования.

БИЛЕТ N 7

1. Общий подход к делению проектирования.

Этапы проектирования:

1. Деление проектирования по времени выполнения работы.

- Научно-исследовательские работы. Результат – технические предложения на использование полученных результатов
- Опытно-конструкторские работы. Результат – эскиз будущего устройства (выбирается лучший)
- Техническое (рабочее) проектирование. Результат – конкретная тех. документация на изготовление устройства (принципиальные схемы со всеми описаниями).
- Производство опытного образца
- Испытания опытного образца

2. Вертикальное деление (по характеру учитываемых свойств объекта). 4 этапа:

- функционального проектирования (анализ ТЗ и корректировка, разработка структурных, функциональных, логических и принципиальных схем пр. объекта)
- алгоритмическое пр-е (все что связано с ПО – от системы команд до ОС)
- конструкторское пр-е (документация для изгот. изделия – все что связано с физич. Реализацией)
- технологическое пр-е (вкл. Разработку док-ции на технологию изгот-ния проектируемого объекта)

3. Горизонтальное деление .

Каждый из этапов вертикального деления делятся также по горизонтали:

ФУНКЦИОНАЛЬНОЕ ПР - НИЕ

1. Системное проектирование; 2. Логическое проектирование; 3. Схемотехническое проектирование; 4. Компонентное проектирование.

АЛГОРИТМИЧЕСКОЕ ПР - НИЕ

1. Программирование всей системы; 2. Программирование модулей; 3. Проектирование микропрограмм.

КОНСТРУКТОРСКОЕ ПР - НИЕ

1. Проектирование "шкаф-стойка"; 2. Проектирование панелей; 3. Проектирование ТЭЗ (технический элемент замены); 4. Проектирование модуль, кристалл, ячейка.

ТЕХНОЛОГИЧЕСКОЕ ПР - НИЕ

1.Разработка принципиальных схем технологического процесса;
2. Маршрутная технология; 3.Разработка технологических операций.

2. Алгоритм ускоренной итерации (2-ая модель сигналов).

Количество тактов (проходов) схемы (максимальное количество итераций) = количество обратных связей + 2.

При использовании данного алгоритма максимальное число итераций в последовательной схеме из 3 элементов, при изменении входного набора, будет равно 2 тактам, а не за 4 как в алгоритме простой итерации.

Для данного алгоритма:

$$Y_{ij}^{k} = F\left(x^{k}, y_{i1}, ..., y_{ij-1}, y_{i-1j}, ..., y_{i-1n}\right).$$

Если нумерация элементов в схеме неправильная нужно перенумеровать элементы.

Необходимо определить начальное состояние схемы. При вычислении значения на выходе i – того элемента на итерации h значения на входах выбираются следующим образом: если входная линия (линиями схемы будем называть внешние входы схемы и выходы элементов) имеет номер $j < i$, используется ее значение, полученное на итерации h, а если $j > i$, — значение, полученное на итерации h - 1.При использовании итераций Зейделя требуемое число итераций зависит от порядка нумерации линий схемы. Поэтому используют алгоритмы ранжирования.

БИЛЕТ N 8

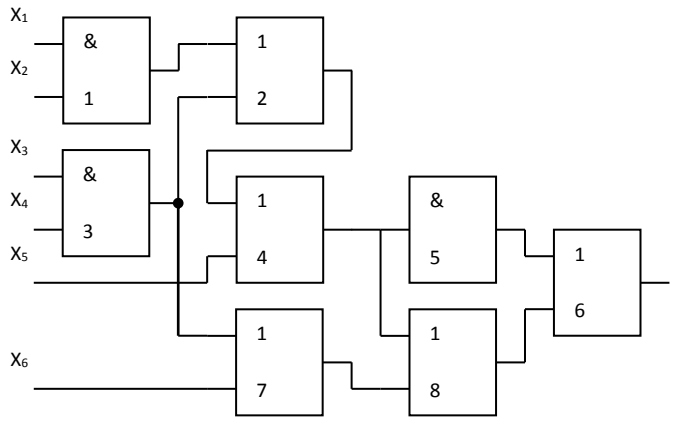
1. Алгоритм ранжирования.

Нулевой ранг-все входные контакты. В следующий ранг записываются элементы, которые имеют связь ***только*** с входными контактами. В i-й ранг записываются элементы, входы которых уже определены. И так далее, пока не будут отранжированы все элементы.

1. Записываем нулевой ранг.
2. Просматриваем всю схему и ищем все элементы i-го ранга.
3. Повторяем все, пока не отранжируем всю схему.

Данный алгоритм не применяется для схем с обратными связями.

Пример.



R_0 – X_1 X_2 X_3 X_4 X_5 X_6

R_1 – 1, 3 (k=2)

R_2 – 2, 7 (k=4)

R_3 – 4 (k=5)

R_4 – 5, 8 (k=7)

R_5 – 6 (k=8)

Данный алгоритм не работает для схемы с обратной связью. Для решения данной проблемы используется алгоритм условного ранжирования.

2. Классификация методов тестового контроля.

Тестовый контроль – специально подобранная тестовая задача, входные воздействия которой позволяет определить исправность объекта. Выполняется в специально отведенные промежутки времени при помощи специально подобранных воздействий.

Тестовый контроль проводится при не функционировании объекта. Все тесты строятся на определении логических неисправностей (0, 1, Not).

В интегральной схеме можно выделить ряд неисправностей, которые отличны от константных неисправностей.

- ближайшее соседство – паразитная связь между ближайшими элементами цифровых устройств. Заключается в топологии размещения элементов на подложке ИС.
- Соседство – топологически не рядом размещенные элементы, паразитическая связь между ними.
- Паразитная задержка
- Увеличение времени прохождения сигнала через объект
- Чувствительность к двоичным наборам. Неправильное функционирование устройства, вызванное сочетанием двоичных значений набора
- Чувствительность к инструкциям – неисправность, имеющая место в микропроцессорах, вызывающая неправильное функционирование устройства при выполнении определенных команд.
- Насыщение шины.

Обычно тесты строятся для выявления одиночных неисправностей (на одном элементе). Для выявления кратных неисправностей необходимо использовать специальные методы.

Элементарная проверка состоит из двух векторов T = {X, Y} X- вектор входных воздействий, Y – вектор реакции.
Тестовая последовательность содержит множество проверок, позволяющих обнаружить все неисправности в заданном классе.

БИЛЕТ N 9

1. Методы сжатия реакций ОК.

Принятие решения о техническом состоянии ОК основывается на анализе выходной последовательности в матричном представлении $Y = \left[y_{t,i} \right]$. Очевидно, что для хранения матрицы необходимы большие объемы памяти, что объясняет важность методов сжатия выходной последовательности (диагностической информации). Методы, которые используют такое сжатие называют методами компактного тестирования.Поскольку методы сжатия диагностической информации слабо зависят от вида испытательной последовательности, то они могут рассматриваться отдельно.В общем случае, задача сжатия выходной информации связана с выбором кодирующего отображения, которое ставит в соответствие последовательности $W = \langle w_1, w_2, \ldots, w_n \rangle$ код $K = \langle k_1, k_2, \ldots, k_m \rangle$ ($m < n$). Если в качестве последовательности W выбраны строки матрицы Y, то говорят о пространственном, а в случае, когда W соответствуют столбцы Y – о временном сжатии диагностической информации.Поскольку эффект сжатия наблюдается только в случае $m < n$, то, естественно, возникает вопрос о достоверности компактного тестирования, т.е. о вероятности того, что примененный метод сжатия позволяет обнаружить все предполагаемые неисправности ОК. При этом анализируется влияние, которое оказывает неисправность на вид выходной последовательности и то, каким образом это влияние отображается в код K.

2. Событийный алгоритм моделирования (статическая модель элементов).

Значение сигнала на выходе элемента может измениться только в том случае, если изменилось значение сигнала хотя бы на одном его входе. Это обстоятельство лежит в основе алгоритмов событийною моделирования, где обработке подлежат только элементы с изменившимися входными значениями сигналов.

При событийном алгоритме моделирования используются две таблицы:

- таблица текущих событий (ТТС):(хранит номера элементов, которые необходимо просчитать в данный момент времени);
- таблица будущих событий (ТБС):(хранит номера тех элементов, на входах которых произошли события).

Алгоритм событийного моделирования состоит в следующем: Задать в схеме начальное состояние.Задать установочный входной набор и просчитать схему на нем любым итерационным либо событийным алгоритмом, но в таблицу будущих событий записать все элементы схемы.Подать первый входной набор. В таблицу текущих событий записать номера входов, на которых произошли события, а в таблицу будущих событий записать те элементы, которые связаны с этими входами.Таблицу будущих событий переписать в таблицу текущих событий и очистить таблицу будущих событий. Просчитать все элементы из таблицы текущих событий.Последовательно моделируем элементы из таблицы текущих событий, при этом, если сигнал на выходе элемента изменился, то его последователя записываем в таблицу будущих событий. Данный процесс продолжается до тех пор, пока не станет пустой одна из таблиц. При генерации таблицы будут повторятся.

БИЛЕТ N 10

1. Деления процесса проектирования по временному признаку.

1) **Научно-исследовательские работы** (НИИ, НИ отделы). Оцениваются готовые проекты и предлагаются новые методы, компоненты, теории и т.д. Результат – предложение испытаний результатов при проектировании новых изделий. Используются специальные САПР (АСП научных экспериментов). Результаты исследований не являются обязательными для испытаний.
2) **Опытно-конструкторские работы (ОКР)** – формируется и согласовывается тех. Задание на разработку нового изделия и выполняются ОКР. Прорабатываются возможные вар-ты проекта, можно ли реализовать данное ТЗ, задаются элементная база, нет ли необходимости ее разработать. Синтезируется эскиз проекта. Результат – эскиз будущего устройства.
3) **Техническое (рабочее) проектирование** – основной этап проектирования, на котором непосредственно разрабатывается проект. Результат – документация на изготовление изделия (принципиальные схемы со всеми описаниями)
4) **Производство опытного образца**
5) **Испытания опытного образца**. По результатам возможны варианты.

2. Событийный алгоритм моделирования (динамическая модель элементов).

Значение сигнала на выходе элемента может измениться только в том случае, если изменилось значение сигнала хотя бы на одном его входе. Это обстоятельство лежит в основе алгоритмов событийною моделирования, где обработке подлежат только элементы с изменившимися входными значениями сигналов.
При событийном алгоритме моделирования используются две таблицы:

- таблица текущих событий (ТТС):(хранит номера элементов, которые необходимо просчитать в данный момент времени);
- таблица будущих событий (ТБС):(хранит номера тех элементов, на входах которых произошли события).

Алгоритм событийного моделирования состоит в следующем:Задать в схеме начальное состояние.Задать установочный входной набор и просчитать схему на нем любым итерационным либо событийным алгоритмом, но в таблицу будущих событий записать все элементы схемы.Подать первый входной набор. В таблицу текущих событий записать номера входов, на которых произошли события, а в таблицу будущих событий записать те элементы, которые связаны с этими входами.Таблицу будущих событий переписать в таблицу текущих событий и очистить таблицу будущих событий. Просчитать все элементы из таблицы текущих событий.Последовательно моделируем элементы из таблицы текущих событий, при этом, если сигнал на выходе элемента изменился, то его последователя записываем в таблицу будущих событий. Данный процесс продолжается до тех пор, пока не станет пустой одна из таблиц. При генерации таблицы будут повторятся.

БИЛЕТ № 11

Деление процесса проектирования по характеру выполняемых работ.

1) **Функциональное.** Исходные данные – ТЗ и пром. результаты алгоритмического проектирования. Цель – разработка функциональных, принципиальных, структурных схем. Уточняется ТЗ, распределяются функции объекта, которые будут выполняться при помощи технических средств.
2) **Алгоритмическое** – Разрабатываются алгоритмы реализации функций проектирования, определяется о/с для работы, разрабатываются программы и микропрограммы для объекта.
3) **Конструкторское** – Разрабатываются конструкции (железо), которые будут реализовывать данный проект.
4) **Технологическое** – Разрабатываются технологии изготовления данного продукта, определяются маршрутные карты, инструменты для реализации и т.д.

Событийный алгоритм моделирования (ЛИД – модель элемента).
Значение сигнала на выходе элемента может измениться только в том случае, если изменилось значение сигнала хотя бы на одном его входе. Это обстоятельство лежит в основе алгоритмов событийною моделирования, где обработке подлежат только элементы с изменившимися входными значениями сигналов.
При событийном алгоритме моделирования используются две таблицы:
- таблица текущих событий (ТТС):(хранит номера элементов, которые необходимо просчитать в данный момент времени);
- таблица будущих событий (ТБС):(хранит номера тех элементов, на входах которых произошли события).

Алгоритм событийного моделирования состоит в следующем:Задать в схеме начальное состояние.Задать установочный входной набор и просчитать схему на нем любым итерационным либо событийным алгоритмом, но в таблицу будущих событий записать все элементы схемы.Подать первый входной набор. В таблицу текущих событий записать номера входов, на которых произошли события, а в таблицу будущих событий записать те элементы, которые связаны с этими входами.Таблицу будущих событий переписать в таблицу текущих событий и очистить таблицу будущих событий. Просчитать все элементы из таблицы текущих событий.Последовательно моделируем элементы из таблицы текущих событий, при этом, если сигнал на выходе элемента изменился, то его последователя записываем в таблицу будущих событий. Данный процесс продолжается до тех пор, пока не станет пустой одна из таблиц. При генерации таблицы будут повторятся.

БИЛЕТ №12

Принципы системного подхода к процессу проектирования.

Система – набор примитивов, объединенных и взаимосвязанных для выполнения задачи. **Основной принцип системного подхода** – рассмотрение сложных систем по частям, обязательно с учетом их взаимодействий. Включает в себя выявление структуры системы, типизацию связей между ее элементами, определение атрибутов и анализ влияния внешней среды. Конкретизация подхода выявляется другими названиями – блочно-иерархический, структурный, и объектно-ориентированный подход. Под **структурным** понимается синтезирование вариантов системы из компонентов и их оценивание при частичном переборе и предварительном прогнозировании характеристик компонентов. **Блочно-иерархический** подход использует декомпозицию сложных объектов и соответствующие средств их создания на иерархические уровни и аспекты, вводит понятие стиля проектирования (восходящее и нисходящее). **Объектно-ориентированный** подход реализуется при проектировании ПО, разработке информационных систем.
Преимущества:
- Вносит в модели приложений большую структурную определенность, распределяя данные и процедуры между классами объектов
- Сокращает объем спецификации, благодаря введению в описание иерархии объектов и отношения наследования между свойствами объектов разных уровней иерархии.
- Уменьшает возможность искажения данных вследствие ошибочных действий за счет ограничения доступа к определенным категориям данных в объекте.
Для всех подходов характерно следующее:
- Структуризация процесса проектирования, выражая декомпозицией проектных задач и документации, выделение стадий, этапов, процедур
- Итерационный характер проектирования
- Типизация и унификация проектных решений и средств проектирования.

Параллельная реализация итерационных алгоритмов.

Количество тактов(проходов) схемы (максимальное количество итераций) = самый длинный путь+1.
В данном алгоритме значение на любом элементе в схеме считается как функции, которая зависит от входного набора и значений элементов на предыдущей итерации:
$Y_{ij}^{k} = F\left(x^{k}, Y_{i-1}\right)$, где $x = \{x_1, x_2, ..., x_l\}$ - входной набор (множество входных сигналов);
$Y = \{y_1, y_2, ..., y_n\}$ - выходные сигналы схемы; *j*- номер элемента; *i* - номер итерации;
k - номер входного набора ($k = \overline{1..m}$).
При параллельной реализации одновременно моделируют несколько входных наборов. Число одновременно моделируемых наборов зависит от языка программирования и особенностей реализации алгоритма. Ускорение происходит за счет меньшего числа обращений к описанию схемы.

- преимущество: простота реализации, ускорение за счет меньшего числа обращений к описанию схемы;
- недостаток: большое время вычисления, большие вычислительные затраты.

БИЛЕТ №13

Деление процесса проектирования по блочно-иерархическому подходу.

В основе лежит декомпозиция проекта по уровням проектирования, выполняемых последовательно.
Уровни **функционального проектирования**
Системный – определяется общая структура проекта; определяется функциональная и структурная схема, после построения схемы определяется, нужно ли проектировать новые эл-ты. Если да, то 2
Функционально-логический – определяются функциональные и логические элементы будущего устройства, и разрабатываются тестовые для контроля.
Технический – разрабатывается реализация устройства, в частности монтаж; определяется покрытие платы реальными элементами; компоновка и размещение элементов в модулях; трассировка печатной платы, анализируется электрические параметры будущего изделия. Если необходимо выполнить проектирование компонентов, документация передается на компонентный уровень.
Компонентный
Уровни **алгоритмического проектирования (программисты)**
Проектирование алгоритмов
Программирование системы
Программирование модулей
Программирование мета-программ
Уровни **конструкторского проектирования (конструкторы)**
- Шкаф, стойка, панель
- ТЭЗ
- Модуль
- Кристалл
- Ячейка

Уровни **технического проектирования (технологи)**
- принципиальная схема технологического процесса
- маршрутные карты
- технологические операции

Последовательностная реализация событийных алгоритмов.

При событийном алгоритме моделирования используются две таблицы:

- таблица текущих событий (ТТС):(хранит номера элементов, которые необходимо просчитать в данный момент времени);
- таблица будущих событий (ТБС):(хранит номера тех элементов, на входах которых произошли события).

Алгоритм событийного моделирования состоит в следующем:Задать в схеме начальное состояние.Задать установочный входной набор и просчитать схему на нем любым итерационным либо событийным алгоритмом, но в таблицу будущих событий записать все элементы схемы.Подать первый входной набор. В таблицу текущих событий записать номера входов, на которых произошли события, а в таблицу будущих событий записать те элементы, которые связаны с этими входами.Таблицу будущих событий переписать в таблицу текущих событий и очистить таблицу будущих событий. Просчитать все элементы из таблицы текущих событий. Последовательно моделируем элементы из таблицы текущих событий, при этом, если сигнал на выходе элемента изменился, то его последователя записываем в таблицу будущих событий. Данный процесс продолжается до тех пор, пока не станет пустой одна из таблиц. При генерации таблицы будут повторятся.

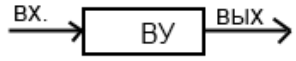
При последовательностном моделировании, моделируется последовательно вся цепочка, без таблиц. Последовательностная реализация является приближенной, время моделирования сокращается на предварительных этапах анализа схем.

БИЛЕТ №14

Блочно-иерархический подход к процессу проектирования.

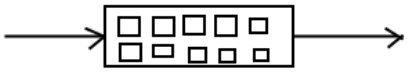
Используется декомпозиция описания сложных объектов и соответствующих средств для их создания на иерархические уровни и аспекты, вводит понятие списка проектирования(восходящее и нисходящее)

На верхнем этапе устройство представляется как черный ящик.



Техническое задание

Системный



Определяется, какие функции, и в каких элементах будут выполняться.

Затем определяется, какие элементы нужно разработать.

С каждым этапом все более высокий уровень детализации.

«Линии отрыва» между уровнями определяются математическим аппаратом, который может быть использован при проектировании на данном этапе.

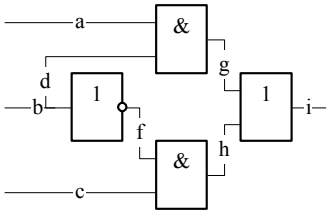
1) На самом верхнем этап используется теория вычислительных систем для решения задачи синтеза и имитационное моделирование для анализа.

2) Синтез – ПТЦА. Анализ – логическое моделирование. Получаем функции и логические схемы.

Синтез и анализ – теория графов, теория множеств.

Метод таблиц истинности синтеза тестов.

Метод таблиц истинности относится к первой группе и, следовательно, использует моделирование ОД с неисправностями. При этом анализируются все из 2^n входных наборов схемы, где n — число входов ОД. После определения классов эквивалентных неисправностей выполняется моделирование, результатом которого является таблица функций различения (ТФР), задающая отношение τ. Затем находится минимальное покрытие этой таблицы. Ниже приведен пример вычисления теста для схемы, изображенной на рис.5.3. Решение задачи определения классов эквивалентных неисправностей представлено в виде табл.5.2.



Набор	abc	i
T_0	000	0
T_1	001	0
T_2	010	0
T_3	011	1
T_4	100	1
T_5	101	1
T_6	110	0
T_7	111	1

	Классы неисправностей									
	1	2	3	4	5	6	7	8	9	10
T_0						1			1	
T_1				1		1	1		1	
T_2		1							1	
T_3	1		1							1
T_4				1	1					1
T_5					1					1
T_6		1	1					1	1	
T_7	1									1

Класс	Неисправность
1	a/0, d/0, g/0
2	a/1
3	b/0
4	b/1
5	c/0, f/0, e/1, h/0
6	c/1
7	d/1
8	e/1, f/1
9	g/1, h/1, i/1
10	[/0

БИЛЕТ №15

Типовые проектные процедуры.

Основные – синтез и анализ. Синтез – непосредственная разработка проекта. Анализ – проверка правильности полученных результатов. Сущность проектирования заключается в принятии проектных решений, обеспечивающих выполнение предъявленных требований. Синтез – основа проектирования. Анализ – вспомогательная, но необходимая процедура.

Синтез:

структурный – разработка структуры будущего объекта

параметрический – определение параметров, реализуемых данным объектом.

Задача параметрического синтеза является задачей оптимизации, если целью является улучшения х-к объекта, либо выбором синтеза наилучшего решения данной задачи.

Постановка задачи структурно синтеза

O = { F, S, P}, O – описание, F – функциональное, S – структурное, P – параметрическое.

Структурный синтез – преобразование функционального и параметрического описания системы в структуру и парам описание элементов.

$\{ F_c, P_c\} \dashrightarrow \{ S_э, P_э\}$

$S_э = \{E, \varphi\}$ E – элементы, φ – связи.

Метод активизации путей синтеза тестов.

Необходимо задать неисправность.

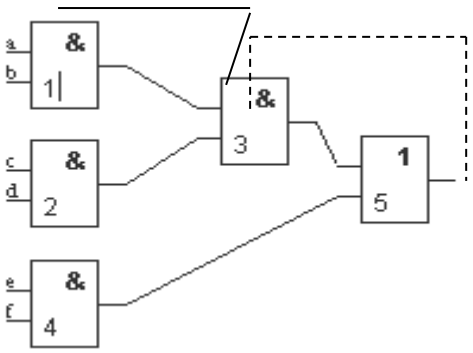
1-я часть – активизация пути

2-я часть – дополнение входного набора

Активизируемый путь – это путь, по которому данная неисправность может быть передана на выход схемы.

a	b	c	d	e	f	1	2	3	4	5
0	X					0(1)		0(1)		0(1)
							1		0	
		1	1	0	X					
							Y			

V



$t_э$=0X110X 0(1)

Если кратные неисправности, то не все можно обнаружить. Если схема с обратными связями, то ее преобразовывают по модели Хаффмана – разрывают ОС (пунктиром) и добавляют вход (V).

$t_э$=0X110X y=1

Установочная последовательность – подается до того, как подается тестовая последовательность. Она устанавливает схему в нужное состояние, но сама тестом не является.

X	V	W
$t_{э1}$	0	1
$_{э2}$	1	1
$t_{э3}$	1	

V- вход, W – выход. Нужно найти такой тест, у которого на входе 1 нужна (W=1, с предыдущего теста).

Сегмент теста – последовательность упорядоченных по изменению внутренних состояний схемы.

Установочных последовательностей нужно столько, сколько будет сегментов.

БИЛЕТ №16

Задача синтеза в процессе проектирования.

Синтез – создание описания объекта, выполняющего заданные функции и удовлетворяющего заданным ограничениям.
Описание – набор инструкций в каком-либо алфавите.
Задача синтеза выполняется в выбранном классе элементарных объектов, из которых составляется объект, реализующий заданный класс функций.
Исходные данные: описание функций, возлагаемых на проектируемый объект; перечень параметров, характериз. качество и ограничения на их значения.
Результат – некоторая структура, реализующая заданный класс функций.
Под СТРУКТУРОЙ объекта понимается множество S = {C,H}, где C – множество элементов, входящих в структуру объекта, а H - множество связей между ними.
Две структуры называются равными, если они реализуют равные функции (F1 = F2), состоят из одинаковых элементов ({C1} = {C2}), которые связаны одинаковыми связями ({H1} = {H2}).
Две структуры называются ЭКВИВАЛЕНТНЫМИ, если F1 = F2, но C1< >C2 и (или) H1< >H2.
Задача синтеза может иметь формальные методы решения – такая задача алгоритмически разрешима, иначе алгоритмически неразрешима. Алгоритмически – неразрешимые задачи решаются в ручную или с помощью эвристических методов (полный перебор).
Различают СИНТЕЗ СТРУКТУРНЫЙ и СИНТЕЗ ПАРАМЕТРИЧЕСКИЙ. Цель структурного синтеза - получение структурных схем объекта, содержащих сведения о составе элементов и способах соединения их между собой. Цель параметрического синтеза – определение числовых значений параметров элементов.
Синтез называется ОПТИМИЗАЦИЕЙ, если определяются наилучшие, в заданном смысле, структуры и значение параметров. Задачу выбора оптимальной структуры называют СТРУКТУРНОЙ ОПТИМИЗАЦИЕЙ.
При расчете оптимальных значений параметров при заданной структуре говорят о ПАРАМЕТРИЧЕСКОЙ ОПТИМИЗАЦИИ.

D-алгоритм синтеза тестов.

D –метод синтеза детерминированных тестов, в основу кот-го положена идея активизации пути. Вычисление тестового набора основано на создании условий проявления неисправности и активизации пути от места ее проявления до выхода схемы. Исп-й в D метод активизации путей предполагает наличие 2-х стадий. На 1-й опред-ся усл-я активизации эл-в, на 2-й осущ-ся выбор таких усл-й для каж-го из эл-в , кот-е были бы непротеворечивы для всей схемы вцелом.D-алгоритм реализует в полном виде метод активизации пути (одномерный путь- один путь транспортировки неисправности)**.** В начале пытаются реализовать одномерные пути, если не получаются, то многомерные.Задаются таблицы истинности, вычисляются все вырожденные покрытия, D-кубы. Манипуляция с таблицами реализует D-алгоритм. В начале пытаются реализовать одномерные пути, если не получаются, то многомерные.
Задаются таблицы истинности, вычисляются все вырожденные покрытия, D-кубы. Манипуляция с таблицами реализует D-алгоритм.
Вырожденные или x-кубы реализуют формулу склеивания ab+ab=a строится с помощью пересечения кубов:
1^1=1; 0^0=0; 1^0=x; 0^1=x . Для синтеза теста ввели добавочный символ в алфавит A={1,0,x,d,$\bar{d}$}
d- символ характеризующий неисправность значений. d={1,0} 1-неисправен ,0-исправен $\bar{d}$ ={0,1} 0-неисправен, 1-исправен.
Правило построения d кубов следующие : 1^1=1; 0^0=0; 1^x=1; 0^x=x; 1^0=d; 0^1=$\bar{d}$; d^$\bar{d}$=0
Каждый d-куб образует правило транспортировки неисправности с соответствующего входа на выход.

БИЛЕТ №17

Задача анализа в процессе проектирования.

АНАЛИЗ - это определение функционального и параметрического описания системы по заданному структурному описанию.
Предмет решения задачи анализа – исследование свойств F, S и P-описаний, полученных на некотором шаге при спуске по дереву проектных решений. Целью такого исследования является оценка качества полученного варианта решения или верификация F-описания на соответствие заданному.
В отличие от задачи синтеза, задача анализа алгоритмически всегда разрешима. Утверждение справедливо, поскольку вариант решения задачи синтеза уже получен и известны, по крайней мере, соответствующие ему F и S –описания.
Задача анализа решается с помощью моделирования.
Наиболее общими методами анализа являются одновариантный (исследование объекта в заданной точке траектории поведения) и многовариантный (исследование свойств объекта в окрестностях заданной точки траектории поведения).
Адекватность – показатель соответствия модели анализируемому объекту.

Метод частной булевой производной синтеза тестов.

Метод булевой производной рассчитан на синтез тестов для одиночных константных неисправностей и использует аналитическую форму функционального описания ОД.

Частной булевой производной называется $\frac{dy}{dx} = y(x_1 ... x_i ... x_n) \oplus y(\bar{x}_1 ... \bar{x}_i ... \bar{x}_n)$ (1)

Так как в булевой $\{0,1\}$

$\frac{dy}{dx} = y(x_1 ... 1 ... x_n) \oplus y(\bar{x}_1 ... 0 ... \bar{x}_n)$ (0)

Выполняется условие проявления неисправности ($\forall$ неисправность – инверсия правильного сигнала.)

Пусть будет существенным $\frac{dy}{dx_i} = 1$ и активизированным условие транспортировки неисправности.

$x_i^{e_i} * \frac{dy}{dx_i} = 1$, (4) где $e_i = \{0,1\}$ $\begin{cases} x_i^{e_i} = x_i, e_i = 1 \\ x_i^{e_i} = \bar{x}_i, e_i = 0 \end{cases}$

$\begin{cases} (3) \\ (4) \end{cases}$ - решая эту систему можно найти все наборы, которые могут быть включены в тесты.

Чтобы получить полный тест необходимо найти производную для всех комбинаций.
$\{x_i, y_i\}, i = \overline{1,n}, j = \overline{1,m}$ для всех одномерных путей.

Правила:

1. $\frac{dy}{dx_i} = \frac{d\bar{y}}{dx_i} = \frac{dy}{d\bar{x}_i} = \frac{d\bar{y}}{d\bar{x}_i}$
2. $\frac{d1}{dx_i} = \frac{d0}{dx_i} = 0, \ \frac{dy_i}{dx_i} = 1$
3. $\frac{d(k_i y(x ... x_n))}{dx_i} = k \frac{dy(x ... x_n)}{dx_i}$
4. $\frac{d(y_1 * y_2)}{dx} = y_1 \frac{dy_2}{dx_i} \oplus y_2 \frac{dy_i}{dx_i} \oplus \frac{dy_1}{dx_i} \oplus \frac{dy_1}{dx_i}$
5. $\frac{d(y_1 \vee y_2)}{dx} = \bar{y}_1 \frac{dy_2}{dx_i} \oplus \bar{y}_2 \frac{dy_i}{dx_i} \oplus \frac{dy_1}{dx_i} \oplus \frac{dy_1}{dx_i}$

6. $\frac{d(y_1 \oplus y_2)}{dx_i} = \frac{dy_1}{dx_i} \oplus \frac{dy_2}{dx_i}$

7. Если $y = x_1 \cdot x_2 \cdot ... x_i \cdot ... x_n$ коньюкция $\frac{dy}{dx_i} = x_1 \cdot x_2 \cdot ... x_i \cdot ... x_n$

8. Если $y = x_1 + x_2 + ... x_i + ... x_n$ дизъюнкция $\frac{dy}{dx_i} = \overline{x_1} \cdot \overline{x_2} \cdot ... \overline{x_i} \cdot ... \overline{x_n}$

9. Если $y = y(x_1, x_2, ... x_i, ... x_n)$

$y_n = y_n(x_1, x_2, ... x_i, ... x_n)$

$$\frac{dy}{dx_i} = \frac{dy}{dy_1} \cdot \frac{dy_1}{dx_i} \oplus \frac{dy}{dy_2} \cdot \frac{dy_2}{dx_i} \oplus ... \oplus \frac{dy}{dy_m} \cdot \frac{dy_m}{dx_i} \oplus \frac{d^2 y}{dy_1 y_2} \oplus ... \oplus \frac{d^m y}{dy_1 dy_2 dy_m} \cdot \frac{dy_1}{dx_i} \cdot \frac{dy_2}{dx_i} \cdot ... \cdot \frac{dy_m}{dx_i}$$

$y = y_1 \cdot y_2 \cdot y_3$

$y_1 = x_1 \cdot x_2 + x_3$

$y_2 = x_1 \cdot x_3 + \overline{x_2}$

$y_3 = x_2 \cdot x_3$

Тогда мы получим.

$$\frac{dy}{dx_2} = \frac{dy}{dy_1} \cdot \frac{dy_1}{dx_2} \oplus \frac{dy}{dy_2} \cdot \frac{dy_2}{dx_2} \oplus ... \oplus \frac{dy}{dy_m} \cdot \frac{dy_m}{dx_2} \oplus \frac{d^2 y}{dy_1 y_2} \cdot \frac{dy_1}{dx_2} \cdot \frac{dy_2}{dx_2} \oplus ... \oplus \frac{d^2 y dy_2}{dy_1 dy_2 dx_2} \cdot \frac{dy_3}{dx_2} \oplus ... \cdot \frac{d^3 y}{dy_i dy_2 dy_3} \cdot \frac{dy_1}{dx_2} \cdot \frac{dy_2}{dx_2} \cdot \frac{dy_3}{dx_2}$$

$$\frac{dy_1}{dx_2} = \frac{d(x_1 * x_2 * x_m)}{dx} = \overline{x_1} * \overline{x_2} \frac{dx_3}{dx_2} \oplus \overline{x_1} * \frac{dx_1 x_2}{dx_2} = \overline{x_2} * x_1$$

$$\frac{dy_2}{dx_2} = \frac{d(x_1 x_3 + \overline{x_2})}{dx} = x_1 x_3 \frac{d\overline{x_2}}{dx_2} + x_2 \frac{dx_1 x_3}{dx_2} = \overline{x_1}\,\overline{x_3} = \overline{x_1} + \overline{x_3}$$

После всех преобразований $\frac{dy}{dx_2} = x_1 x_3 \equiv 0$

	x_1	x_2	x_3
	1	1	1
$x_2 \equiv 1$	1	0	1

БИЛЕТ №18

Задача оптимизации в процессе проектирования.

Обобщенная постановка задачи оптимизации

Оптимальными считаются те значения, которые удовлетворяют ТЗ и являются лучшими из достижимых.

Задача оптимизации САПР сводится к преобразованию физического представления об объекте, о его назначении и степени полезности в математическую формулировку экстремальной задачи. Цель оптимизации выражается в критериях оптимизации.

Критерии – правила предпочтения сравниваемых вариантов. Основу критериев оптимизации составляет целевая функция F(x), где x – множество управляемых параметров. Векторы x с фиксированными значениями определяют один из вариантов объекта и его характеристики.

Целевая функция должна быть такой, чтобы по ее значениям можно было определить степень достижения цели, т.е. лучший вариант должен характеризоваться большим значением F(X), тогда оптимизация заключается в максимализации F(X) или, наоборот, при минимизации F(X) лучший вариант должен характеризоваться меньшими значениями параметров.

Кроме целевой функции F(X) и перечня управляемых параметров (X) в постановку задачи оптимизации могут входить ОГРАНИЧЕНИЯ ТИПА РАВЕНСТВ H(X) = 0 и НЕРАВЕНСТВ H(X)<>0. Частным случаем ограничений типа неравенств являются прямые ограничения ai<=xi<=bi, где ai и bi - предельно допустимые значения параметра xi.

Область пространства управляемых параметров, в которой выполняются заданные ограничения, называет ДОПУСТИМОЙ ОБЛАСТЬЮ XD.

Объект называется строго оптимальным, если значения всех параметров находятся в допустимой области значений параметров.

Объект называется квазиоптимальным, если некоторые параметры из вектора (X) выходят за границы ограничений, но при этом ограничения границ должны быть строго заданы.

При наличии ограничений задача оптимизации называется УСЛОВНОЙ ОПТИМИЗАЦИИ, в противном случае – БЕЗУСЛОВНОЙ.

Область, в которой выполняются как прямые ограничения, так и условия работоспособности, называется ОБЛАСТЬЮ РАБОТОСПОСОБНОСТИ.

Таким образом, итоговая формулировка задачи оптимизации при проектировании имеет вид: экстремизировать целевую функцию F(X) в области XD, заданной ограничениями H(X) = 0 и ф(X) > 0.

Задача оптимизации в такой постановке есть задача математического программирования. При линейности функций F(X), H(X), ф(X) - задача линейного программирования. Если хотя бы одна из них нелинейная - задача нелинейного программирования..

Если все (или часть) X - дискретны, то задача дискретного (или частично дискретного) программирования.

Дискретное программирование называется целочисленным, если X принадлежит множеству целых чисел. Если XD есть пространство булевых переменных, то – задача бивалентного программирования.

Задача структурной оптимизации сводится к построению оптимальной структуры S = (E,H). При этом под ОПТИМАЛЬНЫМ будем понимать такой вариант структуры, параметры которой удовлетворяют всем системным, конструктивным, технологическим, электрическим и экономическим требованиям ТЗ, а критерий оптимальности, описывающий качество проектируемой структуры, принимает экстремальное значение.

Цепной метод поиска булевой производной.

$$\frac{dy}{dx_i} = y(x_1,...,x_{i-1},0,x_{i+1},...,x_n) \oplus y(x_1,...,x_{i-1},1,x_{i+1},...,x_n)$$

Выполнив суперпозицию можна записать

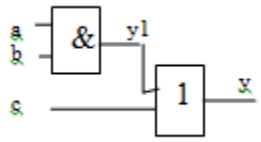
$$Y = y(x_1,...,x_{k-1} * y_1(x_k,...,x_i,...,x_n))$$

$$\frac{dy}{dx_i} = \frac{dy}{dy_1} \cdot \frac{dy_1}{dx_i}$$

$$\frac{dy}{dx_i} = \frac{dy}{dy_1} \cdot \frac{dy_1}{dy_2} \cdot \ldots\ldots \cdot \frac{dy_m}{dx_i}$$

Идея метода – при разумном выборе функции $y_1 \ldots y_m$ все производные в правой части берутся достаточно просто. Но это справедливо в том случае, если при каждой суперпозиции находится единственная ф-ция $y_1 \ldots y_m$ которая зависит от x_i. Это соответствует комбинационной схеме без разветвлений.

Пример



y = ab v c
y1 = ab

$$\frac{dy}{db} = \frac{dy}{dy_1} * \frac{dy_1}{db}$$

$$\frac{dy}{dy_1} = \frac{d(ab \vee c)}{d(ab)} \overset{(8)}{=} \bar{c}$$

$$\frac{dy_1}{db} = \frac{d(ab)}{db} = a$$

$$\frac{dy}{db} = a\bar{c}$$

	a	b	c	y
b=0	1	1	0	1
b=1	1	0	0	0

Вычисления можна упростить используя префиксную форму

$$y = (((x1 + x2)_1 x3x4)_3 (x5 + x6)_2)_4$$

$$y = (_4 \wedge (_3 \wedge (_1 \vee (x1x2)_1 x3x4)_3 (_2 \vee x5x6)_2)_4$$

Аргументы в скобках называются списком аргументов

Правило интерпретируется

$$\frac{d(\overline{\wedge} список\_аргументов)}{d(аргумент\_из\_списка)} = \wedge список\_аргументов\_без\_аргументаив\_списка$$

$$\frac{d(\overline{\vee} список\_аргументов)}{d(аргумент\_из\_списка)} = \overline{\wedge} список\_аргументов\_без\_аргументаив\_списка$$

БИЛЕТ №19

Математическая постановка задачи оптимизации.

Оптимальный – удовлетворяющий целевую функцию и укладывающийся в имеющиеся ресурсы. Постановка задачи: следует преобразовать физическое представления о назначении и степени полезности объекта в математическую формулировку экстремальной задачи. Цель оптимизации выражается в критериях оптимизации. Основа критерия – целевая функция F(x), где x – множество управляющих параметров. Фиксация значений вектора управляемых параметров представляет некоторое решение задачи оптимизации. Как правило, в вектор входят все параметры, которые характеризуют объект, либо часть их, тогда остальные либо фиксированы, либо заданы областями. Следовательно, на часть параметров накладывается некоторые ограничения. Ограничения задаются математически в виде неравенств либо равенств, либо прямые ограничения. Задача оптимизации называется задачей условной оптимизации, если все ограничения заданы неравенствами. Безусловной – равенствами. Область параметров, кот. удовлетворяют области ограничений, наз. допустимой областью значений. XD

В области допустимых значений параметров экстремум f(x)

$$\underset{x\in xD}{extr}\, F\left(x\right),\ \textit{где}\ \mathrm{xD}=\left\{\mathrm{x}|,\ \varphi\left(\mathrm{x}\right)=0,\ \psi\left(\mathrm{x}\right)\le 0\right.$$

Задача оптимизации в такой постановке – это задача мат. программирования. Если все функции линейны – задача линейна. Если хоть одна нелинейна – нелинейна

Если все или часть параметров дискретны – соответствующие дискретны x є Z, или частично дискретны – целочисленное программирование. Если x є {0,1} – бивалентное программирование. При структурной оптимизации необходимо найти оптимальный вариант – множество элементов и связей между ними.

Оптимальная – структура, параметры которой удовлетворяют всем системным. конструктивным. технологическим и эконом. требованиям ТЗ, а критерий оптимальности принимает экстремальное значение. Следовательно, в формализованном виде задача структурной оптимизации заключается в определении значений независимых переменных Xi при которой критерий оптимальности F(x) есть множество независимых переменных принимается экстремальное значение.

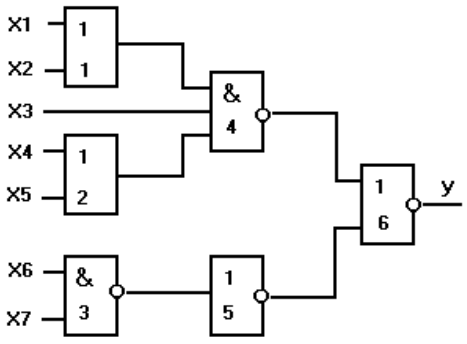
Применение префиксной формы задания функции в цепном методе поиска булевой производной.

Обозначения: **Λ**–конъюнкция(**<u>Λ</u>**–инверсная), **V**–дизъюнкция(**<u>V</u>**–инверсная), **<u>x</u>**-инверсное значение. Поиск булевой производной упрощается, если вместо обычной скобочной формы использовать префиксную форму записи. **y**=x1x2+x3=(V(Λx1x2)x3). Список аргументов – аргументы в скобках. Правило 7 и 8 преобразовывается так:

7. d(<u>Λ</u>(список аргументов))/d(аргумент из списка)=Λ(список без аргумента).
8. d(<u>V</u>(список аргументов))/d(аргумент из списка)=<u>Λ</u>(список аргументов без данного аргумента).

Чтобы найти значение **y** достаточно проинвертировать значение **x**i столько раз, сколько инверсий встречается в активизированном пути.

<u>Пример</u>



y6 = (<u>V</u>(<u>Λ</u>(Vx1x2)x3(Vx4x5))(<u>V</u>(<u>Λ</u>x6x7))) получился активизированный путь.

dy/dx1 = (dy6/dy4)·(dy4/dy1)·(dy1/dx1).

dy6/dy4 = d((<u>V</u>(<u>Λ</u>(Vx1x2)x3(Vx4x5))(<u>V</u>(<u>Λ</u>x6x7)))/d((<u>Λ</u>(Vx1x2)x3(Vx4x5))).

dy4/dy1 = ... = x3(Vx4x5). dy1/dx1 = <u>x2</u>.

dy/dx3=x6x7·x3·(xV4x5)·x2=x6x7·x3x2x4Vx6x7·x3x2x5=x6x3x2·x4Vx7x3x2x4Vx6x3x2x5Vx7x3x2·x5 Vx7x3x2x5.

x1	x2	x3	x4	x5	x6	x7	y
1	0	1	1	x	0	x	1
1	0	1	1	x	x	0	1
1	0	1	x	1	0	x	1
1	0	1	x	1	x	0	1

dy6/dy4=Λy5. dy4/dy1=Λx3y2. dy1/dx1=Λx2. Если обратная связь → разорвать, сделать новый вход, найти производную и установочную последовательность.

БИЛЕТ №20

Общая характеристика критериев оптимизации.

Критерий оптимальности – это правильно предпочтений сравниваемых вариантов. Критерий, который характеризует весь объект называется частным критерием. Каждый критерий можно назвать частным, если он не взаимодействует со всеми остальными.

Если в функции можно задать характеристики всех необходимых критериев, такой критерий называется обобщенным. В качестве обобщенных критериев наиболее часто используется аддитивный, мультипликативный, минимаксный. Если критерий не учитывает вероятностный разброс параметров – он детерминированный, иначе – статистический.

В аддитивных критериях целевая функция образуется путем сложения нормированных значений частных критериев. Нормированные критерии представляют собой отношение реального значения частного критерия к некоторой нормирующей величине, измеряемой в тех же единицах, что и сам критерий (приводит к безразмерной величине).

Возможны несколько подходов к выбору нормирующего делителя.

Первый подход предполагает принимать в качестве нормирующих делителей максимальных значений критериев, достигаемых в области существующих проектных решений.

Второй подход предполагает принимать в качестве нормирующих делителей то оптимальное значение, которое задано в ТЗ.

Третий подход предполагает в качестве нормирующих делителей использовать разность между максимальным и минимальным значением критерия в области компромисса.

Целевая функция: $F(x)=\sum_{i=1}^{n}\frac{F_i(x)}{F_i^H(x)}=\sum_{i=1}^{n}f_i(x)$

$F_i^H(x)$ – нормирующий делитель от частного коэффициента

Необходимо учитывать важность параметров, добавим коэффициент: $F(x)=\sum_{i=1}^{n}c_i f_i(x)$

Недостатки: формальный прием, не вытекает из объективной роли частного критерия; может происходить взаимная компенсация частных критериев.

Мультипликативный критерий. Иногда, важно учитывать не абсолютное значение критерия, а его изменение при решении некоторой задачи.

Целевая функция: $F_i(x)=\prod_{i=1}^{n}F_i(x)$

В случае неравноценности частных критериев необходимо ввести весовой коэффициент Ci и тогда мультипликативный критерий примет вид: $F_i(x)=\prod_{i=1}^{n}C_iF_i(x)$

Достоинством мультипликативного критерия является то, что при его использовании не требуется нормирования частных критериев.

Недостаток: критерий может компенсировать чрезмерные изменения одних критериев за счет изменения других.

Минимаксный критерий.

Формально принцип максмина формулируется следующим образом:

Необходимо выбирать такое множество X0 е X, на котором реализуется максимум из минимальных значений частных критериев F(x0) = max min { fi(x) }.

Если частные критерии fi(x) следует минимизировать, то используется принцип минимакса F(x0) = min max {fi(x)}.

Аддитивные критерии выбирают, когда существенное значение имеют абсолютные числовые значения критериев при выбранном векторе X.

Метод эквивалентных нормальных форм синтеза тестов.

Этот метод основан на представлении булевой функции и виде эквивалентной нормальной формы (ЭНФ), описывающей конкретную реализацию схемы. Поскольку ЭНФ представляет собой сумму логических произведений, она соответствует гипотетической схеме нескольких И-ИЛИ. Каждой схеме И соответствует один терм ЭНФ. Из такого представления ЭНФ становится очевидным, что

для выявления неисправностей, связанных с переменной x_i, входящей в какой-либо терм ЭНФ, необходимо выполнение следующих условий:

1. равенство нулю всех термов, кроме содержащего переменную x_i;
2. равенство единице всех переменных терма, в который входит тестируемая переменная x_i.

Выполнение этих условий обеспечивает тождественное равенство $f(x)=x_i$ и, как следствие этого, выявление неисправностей, связанных с этой переменной, так как неисправность переменной приведет к изменению сигнала на выходе схемы.

Эквивалентная нормальная форма, как и обычная нормальная, вычисляется методом подстановки, с той лишь разницей, что избыточные термы не исключаются, так как они характеризуют конкретную реализацию схемы.

При построении тестов важно не только обеспечить проверку входных переменных, но и всех путей, т.е. необходимо обеспечить проверку одной и той же переменной в разных термах, которым соответствуют разные пути в схеме.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 21

1. Общая характеристика уровня функционально-логического проектирования.

Модель ОП на функционально-логическом уровне проектирования обычно формулируется в терминах структурной теории автоматов. Логический элемент (автомат) может задаваться двумя способами.

Первый способ предполагает, что для описания объекта, имеющего внутренний алфавит *Q*, входные и выходные полюса x_i, i = 1,n иy_j, j = 1,m соответственно, необходимо:

• чтобы каждому входному полюсу x_i, i = 1,n был приписан один и тот же входной алфавит *X*, аналогичным образом, каждому y_j, j = 1,m - выходной алфавит *Y*;

• задать систему канонических уравнений

$$\begin{cases} y_1(t) = \Phi_1(x_1(t), \ldots, x_n(t), q(t)), \\ \ldots\ldots\ldots\ldots \\ y_m(t) = \Phi_m(x_1(t), \ldots, x_n(t), q(t)), \\ q(t+1) = \Psi(x_1(t), \ldots, x_n(t), q(t)). \end{cases} \quad (4.1).$$

В соответствии со *вторым подходом* детерминированный элемент может быть задан как пятерка <X, Y, Q, Ф, Ψ> , где

- *X, Y, Q*- соответственно входной, выходной и внутренний алфавиты;
- Ф- функция выходов
- Ψ- функция переходов

Различают два вида схем: комбинационной принято называть схему, реализующую некоторый тривиальный оператор (набор булевых функций), а последовательностной - схему, реализующую некоторый автоматный оператор.

2.Синтез тестов для последовательностных схем.

Для последовательностных схем известен метод синтеза установочных последовательностей, предложенный Хенни. В этом методе ОД представляется автоматной моделью, описанной в табличном виде. Следовательно, неисправности, обнаруживаемые этими тестами, относятся к классу A_0 . Вместе с тем верхняя граница длины тестовой последовательности для ОД, имеющего n состояний, составляет величину n!, что абсолютно неприемлемо с практической точки зрения.

При тесте последовательностных схем, последовательность должна выстраиваться по последовательности изменений внутренних состояний схемы. Перед поданием тестовой последовательности необходимо установить схему в нужное состояние с помощью установочной последовательности. Установочная посл.+ тестовая = сегмент теста.

Второй подход связан с моделированием ОД. Наряду с достоинством, которое заключается в том, что методы этой группы позволяют синтезировать тесты для довольно больших схем, имеется ряд недостатков, связанных с тем, что методы этой группы не гарантируют получения оптимальных, и даже полных, тестовых последовательностей.

Таким образом, в настоящее время по прежнему актуальны исследования, связанные с разработкой методов тестирования рассматриваемых ОД. Эти исследования развиваются в двух взаимосвязанных направлениях. Первое из них связано с разработкой методов, ориентированных на уровни описание ОД, близкие к системному. Второе - с разработкой специальной методологии тестопригодного проектирования устройств.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 22

1. Задача синтеза на уровне функционально-логического проектирования.

Синтез – создание описания объекта, выполняющего заданные функции и удовлетворяющего заданным ограничениям. Описание – набор инструкций в каком – либо алфавите.
Задача синтеза выполняется в выбранном классе элементарных объектов, из которых составляется объект, реализующий заданный класс функций. Исходные данные: описание функций, возлагаемых на проектируемый объект; перечень параметров, характериз. качество и ограничения на их значения. Результат – некоторая структура, реализующая заданный класс функций.
Под СТРУКТУРОЙ объекта понимается множество S = {C,H}, где C – множество элементов, входящих в структуру объекта, а H – множество связей между ними. Синтез называется ОПТИМИЗАЦИЕЙ, если определяются наилучшие, в заданном смысле, структуры и значение параметров. Задачу выбора оптимальной структуры называют СТРУКТУРНОЙ ОПТИМИЗАЦИЕЙ. Формулировка: из заданного функционального описания и множества параметров построить структуру с конкретными значениями параметров {F, P0} → {S, P} (F – функция, P0 – ограничения на параметры, S – структура, P - параметры).
Этапы:

- системный (теория систем) – решается вручную;
- функционально – логический (теория цифровых автоматов) – на этом этапе используются существующие четкие формальные способы и методы;
- конструкторский (теория графов и множеств) – задача синтеза наиболее формализована.

2. Спектральные коэффициенты.

Сжатие информации с помощью спектральных коэффициентов используется при исчерпывающем компактном тестировании комбинационных схем.Пусть на вход схемы поступает набор $x_1,x_2,...,x_n$, $i_1,i_2,...,i_t$ – номера тех разрядов входного набора функции $F(x_1,x_2,...,x_n)$, зависимость F от которых необходимо определить [50]. *Функциями Уолша* w_i называются функции, принимающие значения ±1 и вычисляемые в соответствии с формулой: $w_{i_1,i_2,...,i_l}(x)=\prod_{j=1}^{l}R_{i_j}(x)$, где $R_{i_j}(x)$ - *функция Радемахера:* $R_{i_j}(x)=(-1)^{x_{i_j}}$. Всего имеется 2^n функций Уолша. Например, для функции $F(x_1,x_2,x_3)$ имеется восемь функций Уолша: w_0, w_1, w_2, w_{12}, w_3, w_{13}, w_{23}, w_{123}.Спектральным коэффициентом или коэффициентом Уолша называется функция $S(i_1,...,i_l)=\sum_{x=0}^{2^m-1}F(x)\,w_{i_1,i_2,...,i_l}(x)$, показывающая меру зависимости значения функции от суммы по модулю 2 разрядов $x_{i_1},...,x_{i_l}$ входного набора; $S(i_j)$ – зависимость от i_j-го разряда.Коэффициенты Уолша можно вычислить также по следующей формуле: $S=T_n\cdot F$ (3.2), где T_n есть $2^n\times2^n$ – трансформационная матрица, определяемая следующим образом: $T_n=\begin{vmatrix}T_{n-1} & T_{n-1}\\ T_{n-1} & -T_{n-1}\end{vmatrix}$,а T_0=[l]. Строки матрицы T_n представляют собой значения функций Уолша.
На рис.3.6 показана схема тестирования со сжатием результатов с помощью спектральных коэффициентов. Генератор функций Уолша, показанный на этом рисунке, управляет выбором проверяемого спектрального коэффициента. Накопитель коэффициента представляет собой простой реверсивный счетчик. Схема генератора для функций от четырех переменных приведена на рис.3.7. C_i предназначено для выбора генерируемой функции Уолша. Генератор для коэффициентов $s_0,s_1,...,s_n$ тривиален. Для s_0 требуется, чтобы все константы c_i были равны нулю, а для $s_1,s_2,...,s_n$ требуется простой мультиплексор.

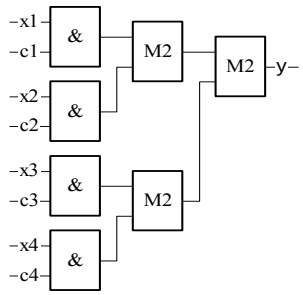


Рис.3.7.

c_1	c_2	c_3	c_4	y
1	0	0	0	x_1
0	1	0	0	x_2
0	0	1	0	x_3
0	0	0	1	x_4
1	1	0	0	$x_1 \oplus x_2$
1	0	1	0	$x_1 \oplus x_3$
...	...	...	...	
1	1	1	1	$x_1 \oplus x_2 \oplus x_3 \oplus x_4$

Для подсчета каждого коэффициента требуется полный прогон тестовых наборов. Распространена и другая интерпретация спектральных коэффициентов, при которой логическому 0(1) ставится в соответствие число –1(+1). Каждый спектральный коэффициент вычисляется умножением значения функции (+1 или -1) на соответствующее значение (+1 или -1) функций Уолша и суммированием произведений по двум наборам входных переменных. При этом значение каждого коэффициента лежит в пределах от —2^n до +2^n.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 23

1. Задача анализа на этапе функционально-логического проектирования.

Задача анализа схемы сводится к двум задачам:
1)статический анализ
2) динамический анализ.
В стат. Анализе исп. Идеальные модели элементов схемы позволяет проверить только корректность процедуры синтеза. При решении задачи статического анализа решаются следующие задачи:

- определения множеств входных сигналов.
- определить достижимости требуемого состояния.
- установление закона функциональной схемы
- определить множества последовательности входных сигналов, вызывающих заданную последовательность внутренних и выходных сигналов.
- сравнение характеристик различных решений.

Динамический анализ определяет характеристики переходных процессов и решает задачи, дополнительные и статическому анализу:

- определение параметров сигнала во время переходного процесса.
- анализ частотных характеристик схемы.
- определение алгоритмической устойчивости схемы

Цель задачи анализа: определить функциональность при заданном структурном и параметрических описаниях. Задача анализа на функционально-логическом уровне использует аппарат передаточных функций для непрерывных моделей и аппарат матлогики и конечных автоматов.

2. Синдром.

Синдром- это один из основных методов сжатия выходной информации.В общем случае, задача сжатия выходной информации связана с выбором кодирующего отображения, которое ставит в соответствие последовательности $W = \langle w_1, w_2, \ldots, w_n \rangle$ код $K = \langle k_1, k_2, \ldots, k_m \rangle$ ($m < n$) . Если в качестве последовательности W выбраны строки матрицы Y, то говорят о пространственном, а в случае, когда W соответствуют столбцы Y – о временном сжатии диагностической информации.Поскольку эффект сжатия наблюдается только в случае $m < n$, то, естественно, возникает вопрос о достоверности компактного тестирования, т.е. о вероятности того, что примененный метод сжатия позволяет обнаружить все предполагаемые неисправности ОК.Синдромное тестирование используется при исчерпывающем компактном тестировании.Синдромом булевой функции называется число $S = K/2^n$, где K – число минтермов функции; n – число входов проверяемой схемы .

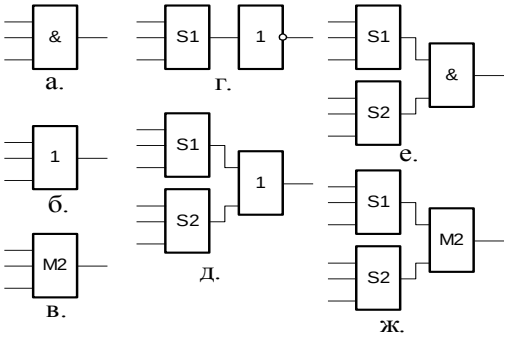
Синдром используется для тестирования комбинационных схем и требует полного перебора входных наборов. Схема называется синдромно-тестируемой, если любая одиночная неисправность меняет синдром. При неразветвленных входах соотношение между входными и выходными синдромами схемы, на выходе которой стоит инвертор, схема ИЛИ, схема И или схема сложения по модулю 2.

$S_1 = 1 - 2^{-2} = 3/4; \quad S_2 = 1 - 2^{-2} = 3/4; \quad S_3 = 2^{-3} = 1/8;$

$S_4 = 1 - (S_2 + S_3 - S_2 S_3) = 7/32;$

$S = S_1 S_4 = 21/128; \quad K = 21.$

Тестовая процедура заключается в подаче на вход схемы всех входных наборов, определении синдрома и сравнении его с эталонным синдромом (т. е. требуется всего один эталон). На проверяемую схему подаются от счетчика все входные наборы. Выход проверяемой схемы соединен со входом счетчика синдрома, который подсчитывает число единиц на выходе проверяемой схемы. После перебора всех выходных наборов производится сравнение полученного и эталонного синдромов.Следует отметить, что единственным различием между синдромом и числом единиц является неявная запятая в регистре (счетчике) синдрома. Если она считается стоящей слева от числа, находящегося в регистре синдрома, то число является синдромом, если справа – то числом единиц. Для уменьшения длины тестов, равной 2^n, комбинационная схема разбивается на подсхемы, спроектированные таким образом, что каждая из них проверяется по своему синдрому.Для реализации синдромного тестирования комбинационные схемы должны проектироваться таким образом, чтобы синдром исправной схемы отличался от неисправной.



а) $S = 2^{-n}$;

б) $S = 1 - 2^{-n}$;

в) $S = ½$;

г) $S = 1 - S_1$;

д) $S = S_1 + S_2 - S_1 S_2$;

е) $S = S_1 S_2$

ж) $S = S_1 + S_2 - 2S_1 S_2$.

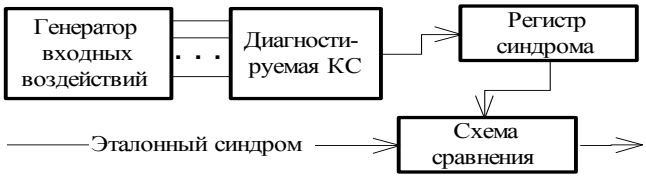


Рис.3.4.

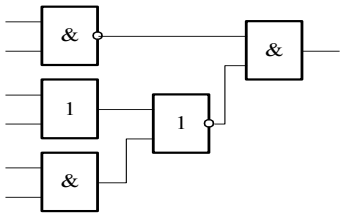


Рис.3.5.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 24

1. Системы логического проектирования.

В системах логического проектирования решаются две задачи: синтеза и анализа;
Синтез – создание описания объекта, выполняющего заданные функции и удовлетворяющего заданным ограничениям.
Описание – набор инструкций в каком – либо алфавите.
Задача синтеза выполняется в выбранном классе элементарных объектов, из кот. составляется объект, реализующий заданный класс функций.
Исх. данные: описание ф-ций, возлагаемых на проектируемый объект; перечень параметров, характериз. качество и ограничения на их значения.
Осуществляется синтез схем и выполняется контроль диагностики этих схем.
Результатом синтеза является функционально-логическая схема, которая потом преобразовывается в принципиальную.
Методы: 1. Теория цифровых автоматов. 2. Абстрактный синтез. 3. Структурный синтез.

Задача анализа - это моделирование. Проще всего использовать натурную модель, но она дорогая.
Математическое моделирование на этом этапе называется логическим.
Методы и алгоритмы для моделирования функциональных и логических схем одинаковы.
Математические модели могут быть аналитическими, имитационными или алгоритмическими.
Методы могут быть: - математическими; - итерационными; - логарифмическими;
Логическое моделирование заключается в построении модели, описывающей его поведение, и определении множества динамически изменяющихся состояний объекта, представляющих определенную реакцию на входные воздействия (результат - таблица реакций). Требование: обеспечить адекватность, как самой модели, так и ее поведения.
Любая задача анализа сводится к *статистическому* и *динамическому анализу*, при котором используются соответствующие модели. Статическая модель - точки времени (пространства), динамическая - изменение состояния на оси времени. Статическая модель идеальна, т.к. не учитывает задержки, инерцию и т.д.

2. Контрольные суммы.

Контрольные суммы- это один из основных методов сжатия выходной информации. В общем случае, задача сжатия выходной информации связана с выбором кодирующего отображения, которое ставит в соответствие последовательности $W=\langle w_1,w_2,...,w_n\rangle$ код $K=\langle k_1,k_2,...,k_m\rangle$ ($m<n$). Если в качестве последовательности W выбраны строки матрицы Y, то говорят о пространственном, а в случае, когда W соответствуют столбцы Y – о временном сжатии диагностической информации. Поскольку эффект сжатия наблюдается только в случае $m<n$, то, естественно, возникает вопрос о достоверности компактного тестирования, т.е. о вероятности того, что примененный метод сжатия позволяет обнаружить все предполагаемые неисправности ОК. **Контрольные суммы.** При использовании контрольных сумм совокупность результатов тестирования рассматривается как массив чисел, над которым выполняется операция поразрядного или арифметического суммирования.Пусть задано упорядоченное множество из *n m*-разрядных чисел $\{u_i\}$, где $i=1,2,...,n$; $u_i=u_{i1},u_{i2},...,u_{im}$ соответствующее выходной последовательности ДУ. Используются следующие способы суммирования:
а) поразрядное суммирование по модулю 2:

$$K_2(u_1,u_1,...,u_n)=v_1,v_2,...,v_m;\quad v_i=\bigoplus_{i=1}^{n}u_{j_i};$$

б) арифметическое суммирование по различным модулям:

$$K_M(u_1,u_1,\dots,u_n)=(u_1+u_2+\ldots+u_n)_{\mathrm{mod}\ M},$$

- $M=n(2^m-1)$ - полная арифметическая сумма;
- $M=2^m$ - арифметическая сумма без учета переноса из старшего разряда;

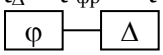
$M=2^m-1$ - арифметическая сумма с циклическим переносом в младший разряд.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 25

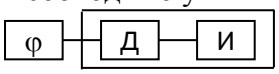
1. Модели элементов в системе логического моделирования.

Элементом называется конструктивно и функционально законченная часть устройства, не подлежащая дальнейшему расщеплению. В общем виде логические элементы описываются E = {φ, A, Δ}, где φ - функция, A – алфавит, Δ - динамические параметры. Обязательно нужно задать φ. Если Δ не задано, то модель – статическая. Если же элемент задан только E = {φ}, то A = {0, 1} – это.
Любой элемент можно представить функциональным и динамическим блоком:
Такая модель предполагает временные характеристики. Если нет временных, то это аналитическая статическая модель (Л -модель).
Самое общее описание задержек – это задержка срабатывания.Δ - блок можно представить в виде задержки срабатывания, если $y(t + t_\Delta) = f(x(t))$. Задержка срабатывания t_Δ предполагает, что модель элемента обладает совершенной задержкой, т.е. временем переключения из одного состояния в другое. Для увеличения адекватности блок Δ может быть расширен с учетом времени фронта и задержки распространения сигнала с входа на выход: $t_\Delta = t_{фр} + t_р$. Т.к. передний и задний фронт отличаются по длительности, то

$t_\Delta = t_{фр}^{01} + t_{фр}^{10} + t_р^{01} + t_р^{10}$.



Это ЛД (логико - динамическая) модель.
Необходимо учитывать инерционные свойства:



Δ- блок

Тогда $t_\Delta = t_{фр}^{01} + t_{фр}^{10} + t_р^{01} + t_р^{10} + t_и^{01} + t_и^{10}$. Это ЛИД – модель.

2. Функции счета.

На одновыходное тестируемое дискретное устройство подается последовательность тестовых наборов *T.* Сдвиговый *m*-разрядный регистр хранит *m* последних результатов $R_i=\{r_i, r_{i+1},..., r_{i+m-1},\}$ [50]. Величина $m\geq 0$ задает число последовательных результатов, подвергаемых анализу с целью определения наличия и числа представляющих интерес признаков сигналов в подпоследовательности R_i последовательности результатов *R.* Анализ осуществляет схема формирования результата, на выходы которой параллельно поступает последовательность результатов $r_i, r_{i+1},..., r_{i+m-1}$, а на выходах формируется текущее значение $C_m(R_i)$ соответствующей функции счета. Сумматор вычисляет текущее суммарное значение $S_m(R_i)$ функции счета путем арифметического сложения $C_m(R_i)$ с хранящимся в регистре накопления результатов предыдущим суммарным значением $S_m(R_{i-1})$ функции счета. Окончательное значение $S_m(R) = \sum_{i=1}^{n-m} C_m(R_i)$ функции счета сравнивается с эталонным значением $S_m(R)_{ЭТ}$ этой функции. Функция счета характеризуется глубиной памяти *m* (числом разрядов *A*) и видом признаков результатов. Наиболее просто реализуется тестирование на основе функций счета, имеющих глубину памяти 0 или 1. Такими функциями являются:
для $m=0$:

а) функция счета единичных значений результатов $S_0^1(R) = \sum_{i=1}^{n} r_i$;

б) функция счета числа переходов изменений значений результатов из 0 в 1 и из 1 в 0

$S_1^2(R) = \sum_{i=2}^{n} (r_{i-1} \oplus r_i)$;

для $m=1$:

в) функция счета числа повторений значений результатов $S_1^3(R) = \sum_{i=2}^{n} \overline{(r_{i-1} \oplus r_i)}$;

г) функция счета числа фронтов (изменений из 0 в 1) $S_1^4(R) = \sum_{i=2}^{n} (\bar{r}_{i-1} r_i)$;

д) функция счета числа срезов (изменений из 1 в 0) $S_1^5(R) = \sum_{i=2}^{n} (r_{i-1} \bar{r}_i)$.

Для компактного тестирования многовыходных ДУ каждому получаемому на выходе ДУ двоичному набору v_i ставится в соответствие его двоичный вес a_i. В качестве функций счета применяются функции

$$S_1^8(V) = \sum_{i=2}^{m} P(a_{i-1} \langle a_i); \quad S_1^9(V) = \sum_{i=2}^{m} P(a_{i-1} \rangle a_i); \quad S_1^{10}(V) = \sum_{i=2}^{m} P(a_{i-1} \neq a_i),$$

где V – последовательность выходных наборов $v_0, v_1,...,v_m$; P – предикат сравнения весов соседних наборов, принимающий значение 1, если результат сравнения совпадает с заданным предикатом, и значение 0 в противном случае.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 26

1. Избыточность и трудоемкость в процедурах контроля и диагностирования СВТ.

Дополнительные затраты, связанные с контрольно-диагностическими мероприятиями, можно характеризовать **избыточностью и трудоемкостью.** Избыточности выделяют три вида: аппаратурную, временную и информационную. Под **трудоемкостью** следует понимать технико-экономический показатель, в общем случае зависимый от перечисленных видов избыточности и учитывающий совокупные затраты на разработку, производство и эксплуатацию средств контроля или диагностики. **Временная избыточность** предполагает дополнительные затраты времени на выполнение контрольных операций. Применительно к тестовому и параметрическому контролю СВТ для выполнения этих операций отводится специальный отрезок времени, т.е. временная избыточность непосредственно не влияет на их производительность. Исключением являются частные случаи контроля, использующие повторное решение задачи с применением тех же самых или эквивалентных алгоритмов и сравнением полученных результатов, поскольку повторное решение может рассматриваться как процедура тестирования, кстати, позволяющая обнаруживать только сбои. При функциональном контроле временная избыточность может оказывать существенное влияние на производительность СВТ. Уменьшить затраты, связанные с временной избыточностью, можно за счет параллельного выполнения основных и контролирующих операций, т.е. аппаратурной избыточности, или использованием для контроля «остатков» времени при синхронной организации вычислений процесса. **Аппаратурная избыточность** имеет место практически всегда (исключение, например, все тот же повторный счет) и определяется необходимостью применения дополнительной аппаратуры для реализации процедур контроля. В случае функционального контроля эта аппаратура «встроена» в объект контроля, в связи с чем такой вид контроля часто называют аппаратным или схемным. **Информационная избыточность** свойственна всем без исключения методам контроля. Так, при тестовом контроле дополнительная информация необходима для хранения образов входных воздействий и эталонных реакций, а при параметрическом, как минимум, образа эталона. Применительно к функциональному контролю информационная избыточность используется в основе большинства методов и заключается в избыточном кодировании информации для проведения контроля и коррекции ошибок в процессе ее преобразований.

2. Частные критерии оптимизации.

Частный критерий – некоторый один параметр, который характеризует качество объекта при ограничениях на все остальные.

$$\underset{x \in xD}{extr}\, F(x),\ где\ \mathrm{xD}=\{\mathrm{x}|,\ \varphi(\mathrm{x})>0,\ \psi(\mathrm{x})=0$$

Если все ограничения заданы только равенством, то функция изменения параметра приводит к изменению всей функции и запасов вариаций задача не имеет.

При проектировании по частным критериям в качестве целевой функции F(X) применяется наиболее важный выходной параметр проектируемого объекта, все остальные параметры в виде соответствующих условий работоспособности относятся к ограничениям. В этом случае задача оптимального проектирования является однокритериальной задачей математического программирования: экстремизировать значение целевой функции F(X) при наличии системы ограничений на параметры проектируемого объекта. Сложность такой задачи небольшая.

Частные критерии выбирают тогда, когда необходимо сравнить несколько эквивалентных решений, либо заранее задана необходимость оптимизации одного или нескольких частных критериев (без существенных ограничений на другие критерии).

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 27

1. Общая характеристика функционального контроля СВТ.

Особое значение в процессе эксплуатации СВТ имеет высокая достоверность определения их технического состояния и своевременное обнаружение отказов и сбоев, поскольку «позднее» их обнаружение может слишком дорого стоить или приводить к неустранимым (катастрофическим) последствиям. Эта задача решается с помощью средств автоматического (встроенного) функционального контроля, которые при появлении ошибки немедленно приостанавливают вычислительный процесс и инициируют работу средств восстановления после сбоя, например повторное выполнение операции. В случае обнаружения отказа средства функционального контроля могут инициировать работу автоматической системы тестового диагностирования, локализующей неисправность. Если рассматривать процесс вычислений как информационный, то все составляющие его операции могут быть разделены на три типа: операции передачи, логические и арифметические преобразования. Следовательно, **для функционального контроля необходимо иметь** средства проверки правильности выполнения всех типов операций и, по возможности, средства коррекции ошибок. Вопросы, связанные с безошибочной передачей информации настолько важны, что их исследование являются предметом самостоятельной теории – теории кодов, контролирующих ошибки. Техническая задача, решаемая с помощью этой теории в общем случае состоит в защите цифровых данных от появляющихся в процессе передачи по каналам связи ошибок. Применительно к функциональному контролю СВТ, независимо от типа операций, в большинстве случаев удобно использовать модель, основанную на применении равномерных избыточных кодов, которые в дальнейшем называются *разделимыми*. Особенностью этих кодов является то, что кодовое слово можно разделить на два подслова фиксированной длины. Первое подслово является самой преобразуемой информацией (в дальнейшем это подслово будет называться кодом информации). Второе – ее кодирующим отображением (контрольным кодом) в соответствующем алфавите, причем это отображение, в общем случае, не взаимно однозначно. С учетом кодирования информации разделимым избыточным кодом при функциональном контроле обычно используется универсальный прием, который сводится к выполнению следующих инструкций:

получить в соответствии с некоторым правилом кодирующее отображение для подслова, являющегося контрольным кодом преобразуемой информации, т.е. сформировать разделимый код;

раздельно выполнить эквивалентные операции преобразования над кодом информации и контрольным кодом;

выполнить повторное формирование контрольного кода применительно к преобразованному коду информации (получить контрольный код результата);

сравнить результат выполнения предыдущего шага с результатом преобразования контрольного кода;

если результат сравнения положителен (отличий в контрольных кодах нет), то полагать, что ошибка отсутствует, иначе инициализировать аварийные действия.

При формализации модели построения контрольных кодов применительно к обработке информации средствами вычислительной техники удобно использовать отображение кода информации в контрольный код, соответствующее представлению чисел в системе счисления в остатках (системе остаточных классов – СОК). Представление чисел в этой системе счисления определяется следующим.

Пусть некоторое целое неотрицательное число A, заданное в позиционной однородной системе счисления, представимо в виде:

$A = aq + r_a$,

где a, q, r_a – целые неотрицательные числа и $r_a < q$.

Теорема 2.1.

Сумма чисел $A_i, i = \overline{1,n}$ *сравнима по модулю* q *с суммой остатков* $|A_i|_q, i = \overline{1,n}$ *этих же чисел:*

$$\sum_{i=1}^{n} A_i \equiv \sum_{i=1}^{n} |A_i|_q .$$

Теорема 2.2.

Произведение чисел $A_i, i = \overline{1,n}$ *сравнимо по модулю* q *с произведением остатков* $|A_i|_q, i = \overline{1,n}$ *этих же чисел:*

$$\prod_{i=1}^{n} A_i \equiv \prod_{i=1}^{n} |A_i|_q$$

Теорема 2.3.

Для заданного множества целых положительных попарно взаимно простых чисел $m_1, m_2, ..., m_k$, и множества неотрицательных целых чисел $c_1, c_2, ..., c_k$, при $c_i < m_i$, система сравнений

$c_i = c \pmod{m_i}$, $i = 1,2, ..., k$,

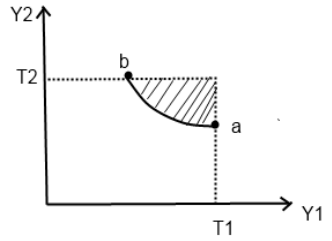
имеет не более одного решения с в интервале $0 \le c \le \prod m_i$, $i = 1,2, ..., k$.

2.Методы задания предпочтений на множестве частных критериев в задаче оптимизации.

Если можно выделить 1 параметр, который наиболее полно описывает объект, то оптимизация ведется по частному признаку, а на остальные накладываются ограничения.
Если все ограничения заданы в виде равенств – задача алгебраическая.
Если все значения параметров заданы в виде равенств и неравенств – задача не имеет степеней свободы.
Частные критерии имеют ряд недостатков: Достаточно часто при выборе экстремального значения 1-го частного критерия, остальные, как правило, даже при наличии ограничений принимают неудовлетворительные значения.

Пример

Пусть необходимо оптимизировать по y1 и y2
y1 <= T1
y2 <= T2



Экстремальное значение параметров будет в точках a и b.Кривая указывает на те точки области работоспособности, которые имеют наилучшие значение параметров.
Множество точек пространства, из которых невозможно перемещение, приводящих к улучшению всех параметров, называется областью Парета.
Частный критерий даету1 как y2. Но тогда при нестабильности работы системы он будет «гулять» в т.ч. вне области.
Тогда берут разницу (Δ) y1<= T1 + Δ, но Δ выбрать сложно.
Поэтому целесообразно брать обобщенный критерий.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 28

1. Риски сбоя в схемах и способы их обнаружения.

Статические состязания: $Y(t_1) = f(x_1)$, $Y(t_2) = f(x_2)$. $Y(t_1) = Y(t_2)$, но в момент времени $t_1<\tau< t_2$ появляется $Y(t_1) \neq Y(t_2)$. В зависимости от того, приведут ли состязания к алгоритмическому переходу различают опасные и неопасные состязания. Опасные состязания приводят к неправильному алгоритмическому переходу. В динамических состязаниях $Y(t_1) \neq Y(t_2)$, но в момент времени $t_1<\tau_1<\tau_2<t_2$ появляется $Y(t_1) \neq Y(\tau_1)$, $Y(\tau_1) \neq Y(\tau_2)$, $Y(\tau_2) \neq Y(t_2)$. Различают функциональные, логические состязания и состязания на уровне логических элементов. Функциональные – если на переходе Y_1 в Y_2 возникает два алгоритмических перехода. Логические – могут дать на выходе неалгоритмический переход. Состязания возникают в логических элементах, выполняющих сложные функции, не за счет гонок, а за счет состязаний внутри элемента.

Для обнаружения статических состязаний в итерационных алгоритмам используют троичную модель элементов. В событийных алгоритмах они обнаруживаются за счет времени между элементами.

2. Общая характеристика параметрического контроля СВТ.

Системы параметрического контроля строятся на основе оценки состояния ОК по некоторым косвенным признакам и обладают неоспоримым преимуществом, выражающимся в не повреждающем характере испытательных воздействий, которые в явном виде, как правило, не используются. При тестовом контроле для доступа к внутренним точкам ОК часто приходится применять специальные игольчатые контактные устройства, которые часто называют ложем из гвоздей (bed of nails), и специальные меры для "отключения" отдельных компонент от остальной части ОК, именуемые применительно к электронным системам «электронными ножами» (electronic knife).Второе преимущество параметрического контроля заключается в следующем. Обычно тесты предназначаются для выявления одиночных неисправностей (под одиночной понимается неисправность с точностью до одного элемента, на выходе которого она может быть зафиксирована). Задача построения тестов для выявления и локализации кратных неисправностей, т. е. одновременно присутствующих в объекте двух или более неисправностей, а также неисправностей, возникающих из-за постепенного ухудшения, например, формы сигналов в настоящее время за исключением простейших ситуаций относится к классу сложных задач диагностики. Контроль и диагностика объектов с такими неисправностями должен производиться по специально разработанным методикам, позволяющим по признакам одиночных неисправностей локализовать группу неправильно функционирующих элементов или цепь с затухающим сигналом. Параметрической контроль в равной мере и без дополнительных затрат позволяет обнаруживать как одиночные, так и групповые неисправности, однако, оказывается ориентированным на класс неисправностей, которые могут устанавливаться только по косвенным признакам, что может быть отнесено к его недостаткам. Например, для электронной аппаратуры отличие температурных или физических полей ОК от эталонных для заданного конструктивного узла не позволяют определить характер самой неисправности и классифицировать ее в рамках, традиционных для этого класса объектов (одиночные константные и инверсные неисправности на входных и выходных полюсах элементов, «ближайшее соседство» – паразитная связь между элементами и т.п.). Кроме того, параметрический контроль в общем случае не позволяет определить неисправности, связанные с динамическими характеристиками объекта («паразитная задержка» –увеличение времени прохождения сигналов через устройство) или его «чувствительности к наборам (инструкциям)» – неправильном функционировании, вызванном определенным сочетанием значений входных переменных (неправильное функционирование программируемых устройств при выполнении определенных команд) и др.Отмеченный недостаток частично компенсируется тем, что параметрический контроль (диагностика) не требует детального структурного описания ОК. Действительно, при решении задач контроля приходится учитывать следующий факт: чем сложнее объект, который необходимо диагностировать, тем меньше информации о его структуре доступно специалисту по диагностированию. Поэтому при создании тестовых программ для таких объектов разработчик вынужден предполагать, что неисправность может привести к реализации любой функции, отличной от заданной. Формализация такого класса неисправностей для тестового контроля достаточно сложна, а для параметрического контроля обычно не вызывает особых затруднений.

Кроме того, параметрический контроль не требует решения задач тестопригодного проектирования ОК, т. е. не требует при проектировании учета таких характеристик объекта, как «управляемость» и «наблюдаемость».

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 29

1. Модели сигналов в системе логического моделирования.

Различают статические и динамические модели. В статических моделях во всех схемах предполагается, что все временные параметры идеальны. В статическом моделировании есть входные параметры, получаем таблицу выходных параметров и можно проверить правильность выполнения функции путем сравнения с ожидаемым.В динамических моделях необходимо учитывать в схеме задержку времени переключения на каждом элементе (Δt – разное), поэтому появляются гонки. В результате появляются неправильные выходные сигналы. Различают статические состязания и динамические.Статические состязания: $Y(t_1) = f(x_1)$, $Y(t_2) = f(x_2)$. $Y(t_1) = Y(t_2)$, но в момент времени $t_1<\tau< t_2$ появляется $Y(t_1) \neq Y(t_2)$. В зависимости от того, приведут ли состязания к алгоритмическому переходу различают опасные и неопасные состязания. Опасные состязания приводят к неправильному алгоритмическому переходу.В динамических состязаниях $Y(t_1) \neq Y(t_2)$, но в момент времени $t_1<\tau_1<\tau_2<t_2$ появляется $Y(t_1) \neq Y(\tau_1)$, $Y(\tau_1) \neq Y(\tau_2)$, $Y(\tau_2) \neq Y(t_2)$. Различают функциональные, логические состязания и состязания на уровне логических элементов. Функциональные – если на переходе Y_1 в Y_2 возникает два алгоритмических перехода. Логические – могут дать на выходе неалгоритмический переход. Состязания возникают в логических элементах, выполняющих сложные функции, не за счет гонок, а за счет состязаний внутри элемента.

Алфавиты:

- $A_2 = \{0, 1\}$. Если он не содержит временные задержки, то это статическая модель. Необходимо расширить алфавит, чтобы проверить динамические характеристики на статической модели.
- $A_3 = \{0, 1, x\}$. x- неопределенное состояние, переходной процесс. Эта модель дает все статические и динамические состязания.
- $A_4 = \{0, 1, \lambda, \varepsilon\}$, где λ - изменение сигнала из 0 в 1, ε - изменение сигнала из 1 в 0.
- $A_5 = \{0, 1, \lambda, \varepsilon, x\}$
- $A_6 = \{0, 1, \lambda, \varepsilon, \delta, x\}$, где δ - статические состязания
- $A_7 = \{0, 1, \lambda, \varepsilon, \delta_1, \delta_2, x\}$, где δ_1 – состязание 0-0, δ_2 – состязание 1-1
- $A_8 = \{0, 1, \lambda, \varepsilon, \delta_1, \delta_2, x, \gamma\}$, где γ - динамические состязания
- $A_9 = \{0, 1, \lambda, \varepsilon, \delta_1, \delta_2, x, \gamma_1, \gamma_2\}$, где γ_1 – состязание 0-0, γ_2 – состязание 1-1

Самый расширенный алфавит – 19 значный. Чем шире алфавит, тем более адекватна модель. С ростом значности алфавита быстро увеличивается время моделирования, а также и требуемый объем памяти.

2.Контроль передачи информации.

Для контроля процесса преобразования информации средствами вычислительной техники, в основном, используются равномерные избыточные коды. Очевидно, что избыточность равномерного разделимого кода определяется длиной подслова, соответствующего контрольному коду. Применительно к СВТ широкого назначения эту избыточность стремяться по возможности минимизировать, даже в ущерб обнаруживающей способности, в силу следующих причин. При формировании избыточного кода можно использовать два приема (иногда оба совмещаются). Первый из них предполагает формирование контрольного кода непосредственно при вводе исходной информации или при получении промежуточных (конечных) результатов вычислений и хранение в памяти всего избыточного кода. Объемы обрабатываемой информации для СВТ сегодня измеряются десятками-сотнями мегабайт и, следовательно, при длине контрольного кода, равном единице, потребуется аналогичное количество физических элементов памяти, высокие надежностные характеристики которых так же нужно обеспечивать. С увеличением длины подслова контрольного кода аппаратурные затраты будут кратно возрастать. Однако, несомненным достоинством рассматриваемого приема является возможность контроля операций записи-чтения – частного случая передачи информации из регистра числа в ячейку памяти и наоборот (по отношению к СВТ вместо «передача» чаще используют термин «пересылка»). Во втором случае информация храниться в памяти в неизбыточном коде. Последний формируется после операции чтения с помощью соответствующего кодера и используется для контроля дальнейших пересылок. При записи информации в память контрольный код игнорируется. Требования к минимизации аппаратурной избыточности в этом случае менее жесткие, но возможность контроля операций записи-чтения теряется. Не менее существенной причиной применения кодов, обнаруживающих только ограниченное множество ошибок, является то, что физическая природа ошибок для аппаратуры СВТ

достаточно изучена и вероятность возникновения кратной ошибки при выполнении операций обработки информации, в отличие от передачи по каналу в системах связи, принимается небольшой. ***Код с проверкой четности***, который широко используется для контроля операции установки равенства или пересылок, образуется путем дополнения информационного подслова контрольным с длиной, равной единице, т.е. одного контрольного разряда. В соответствии с рассматриваемой моделью, информационное подслово здесь рассматривается как код числа в двоичной позиционной однородной системе счисления (в дальнейшем просто двоичной), а контрольный код его отображением в СОК. Такой способ кодирования обычно называют *числовым* и контроль с его применением – *числовым контролем* по модулю. По определению оба подслова есть код в двоичном стандартном алфавите. Следовательно, единственно возможным модулем для представления контрольного кода является число «2». Значение модуля совпадает с основанием двоичной системы счисления, в котором представлено информационное подслово, и обнаруживающая способность контрольного кода оказывается ничтожной, поскольку с его помощью можно установить только ошибку в младшем разряде двоичного числа. Иными словами один контрольный разряд позволяет определить только четное и нечетное значения кода информации, а остальные ошибки обнаружены не будут. Существенно увеличить обнаруживающую способность кода с проверкой четности можно, если изменить вид алфавитного отображения, используя для представления в СОК по модулю «2» *сумму* содержащихся в двоичном представлении информации единиц или нулей. Тогда код позволит обнаруживать все одиночные и кратные нечетным ошибки. Поскольку единицы или нули могут рассматриваться как цифры числового представления, такой контроль часто называют «цифровым контролем по четности». Код Хемминга строится таким образом, что к имеющимся информационным разрядам слова добавляется определенное число контрольных, которые формируются перед передачей путем подсчета четности для определенных групп информационных разрядов.Требуемое число контрольных разрядов определяется следующим образом:
Пусть кодовое слово длиной n разрядов имеет m информационных и $k = n - m$ контрольных разрядов. С помощью k разрядов контрольного слова можно представить 2^k подслов, соответствующих отсутствию или наличию ошибки в информационных разрядах. Таким образом , $2^k \geq n + 1$ или $2^k - k - 1 \geq m$. Контроль с помощью кода Хемминга реализуется с помощью набора схем подсчета четности, которые при кодировании определяют контрольные разряды, а при декодировании – корректирующее слово.Наибольшее применение этот код нашел при контроле в запоминающих устройствах.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 30

1. Модели схем в системе логического моделирования.

В зависимости от способа хранения информации на схеме различают модели: интерпретативные, компилятивные.Компилятивная модель – строится с помощью генерации машинных команд на этапе ввода информации.

При этом различают 2 основных способа построения такой модели:

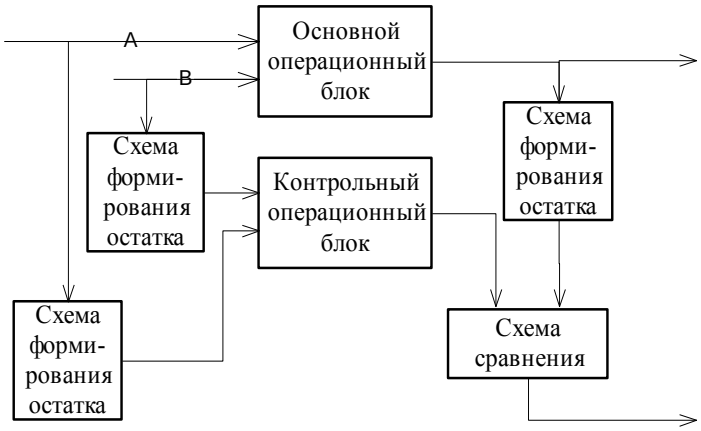
- использование специализированных языков описания схем, которые потом транслируют в машинные коды;
- используются универсальные языки описания. Тогда описание подчиняется синтаксису и семантике языка

Первый способ удобнее, но второй – не требует знаний спец. языков. Интерпретативная модель – представлена в списочном или табличном представлении структур схемы. При этом интерпр. программа использует адреса ее указателей, определяющие связь между элементами. Способ задания схемы никак не влияет на адекватность моделирования.Вычисление значения логическогосигнала на выходе логического элемента сводится к извлечению из списков и их обработки. Преимущество компилятивной – больше быстродействие. Недостаток – малая гибкость т.к. при вводе изменений в структуру схемы необходима повторная компиляция модели, тогда как для интерпретативной - просто вносим в списке указатели.

2. Контроль арифметических и логических операций.

Для контроля арифметических операций обычно используется числовой контроль по модулю q.

Возможность проверки правильного результата выполнения операций сложения и умножения с помощью контрольного кода, представленного в СОК, вытекает из теорем 2.1 и 2.2. При контроле операции деления используется зависимость вида $a = c \cdot b + r$, где a – делимое, c – частное, b – делитель и r – остаток от деления, т.е. в процедуре контроля сравнивается контрольный код делимого с результатом вычисления правой части выражения, содержащего только допустимые в СОК операции.



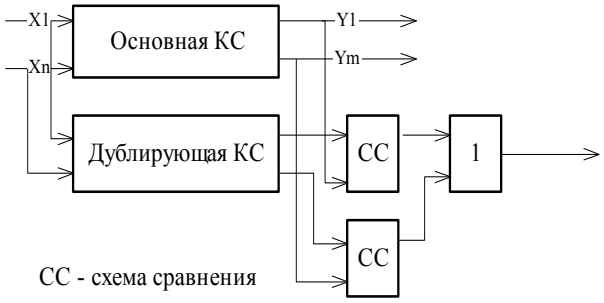
На рис.2.2. представлена структурная схема операционного блока с числовым контролем по модулю (схему можно модифицировать с учетом контроля операции деления, что предлагается сделать в качестве упражнения). Ее аппаратурная избыточность определяется схемами формирования контрольного кода и контролирующего операционного блока, сложность которых зависит от значения q. Этим же значением определяется полнота контроля. При увеличении значения модуля увеличивается число кратных ошибок, обнаруживаемых системой контроля, однако, при этом возрастает и сложность кодирующей и контролирующей аппаратуры. Например, в качестве модуля q не следует выбирать основание системы счисления, используемой для представления основной информации, если она является однородной и позиционной. Это приводит к тому, что контролируются только младшие разряды чисел (последнее уже отмечалось). Избыточность будет минимальной при условии $q = r \pm 1$, где r – основание системы счисления, т.е. для СВТ значение q принимается равным «3». Ограничения на класс обнаруживаемых ошибок для СВТ широкого

назначения при этом считаются приемлемыми . Несмотря на то, что аналоги теорем для суммы цифр операндов и результата сложения и умножения отсутствуют, для контроля арифметических операций можно использовать цифровой контроль, что поясняется следующим примером.Пусть контрольным кодом является остаток по модулю 2 суммы единиц в каждом из операндов. При сложении чисел a и b разряды суммы S образуются следующим образом:

$$\begin{cases} S_1 = a_1 \oplus b_1 \oplus P_1; \\ S_2 = a_2 \oplus b_2 \oplus P_2; \\ \dots \\ S_n = a_n \oplus b_n \oplus P_n. \end{cases}$$

где $S_i, a_i, b_i, P_i, i = \overline{1,n}$ – соответственно разряды суммы, операндов и переноса; $\oplus$ – операция сложения по модулю 2.Тогда, сложив все n равенств по модулю 2 $|S|_2^c = |a|_2^c \oplus |b|_2^c \oplus |P|_2^c$,т.е. при правильном результате сложения четность суммы его цифр должна совпадать со значением, вычисленным согласно правой части приведенного выражения. Применение такого контроля может быть оправдано специализацией СВТ, например, в случае, когда в системе команд предусмотрена только операция сложения и для контроля пересылок используется цифровой контроль по модулю 2.

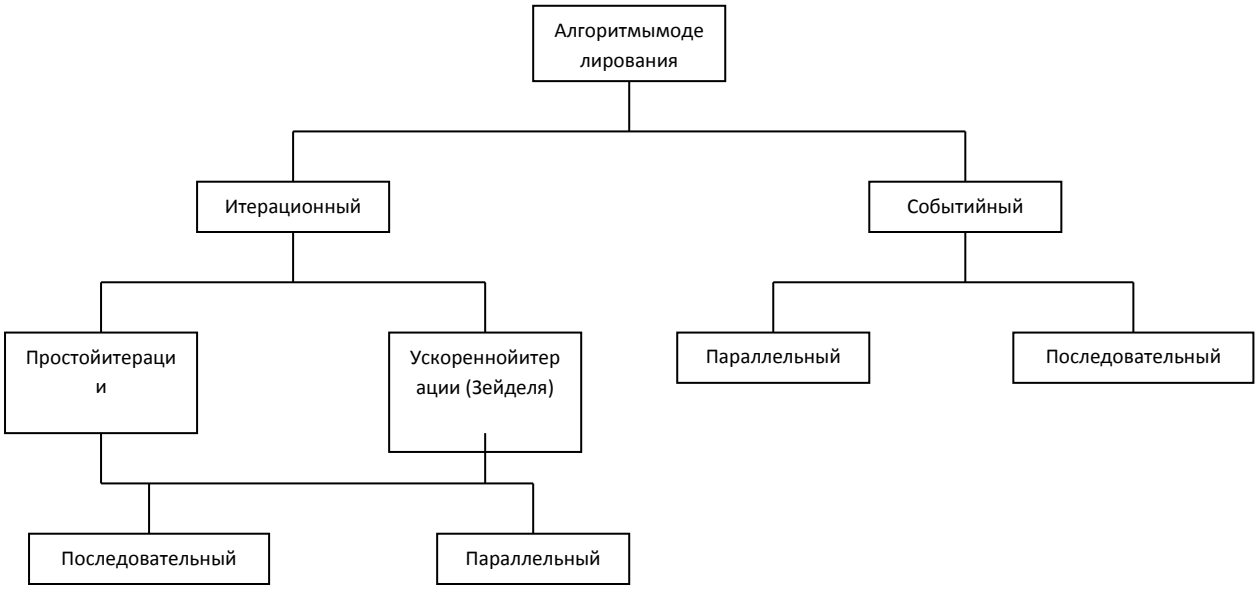
Контроль логических операций.К логическим принято относить операции сдвига, а также операции and, or, not, xor. Последние, начиная с and, могут применяться либо к так называемым логическим значениям true (истина) и false (ложь), кодами которых при n-разрядном представлении являются единица в младшем разряде при остальных нулях и все нули соответственно, либо ко всем разрядам произвольных операндов поразрядно.Модели, с помощью которых могли бы строиться избыточные коды для контроля логических операций, вообще говоря, отсутствуют. Для теории кодов, исправляющих ошибки, такая задача (впрочем, как и контроль арифметических операций) не является предметом анализа. Применение используемой выше модели, точнее построение контрольного кода как отображения в СОК, жестко ограничено теоремами 2.1 и 2.2 (здесь «жестко ограничено» означает, что, например, для контроля сдвига такая возможность есть, поскольку операция сдвига – это умножение на p^n , где p – основание системы счисления, а n – количество разрядов, на которое сдвигается код информации). Перечень подобных «экзотических» приемов можно продолжить, рассматривая операцию *xor* как поразрядную сумму операндов по модулю «2» без учета переносов и корректируя в соответствии с этим выражение (2.1), что, однако, не применимо к другим операциям, и т.д.Единственным очевидным обобщением может быть рекомендация строить избыточный код как разделимый. В этом случае всегда остается возможность использовать в качестве контрольного кода копию кода информации, т.е. дублировать информацию (собственно, представление контрольного кода в виде отображения ее в СОК – это то же дублирование, но не полное и со «сжатием» копии).



ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ N 31

1. Общая характеристика алгоритмов моделирования.

Существует много различных методов и алгоритмов моделирования.
Основными характеристиками алгоритмов моделирования являются адекватность, быстродействие и объем памяти, необходимый при реализации. Под адекватностью понимается степень соответствия результатов моделирования истинному поведению исследуемого ДУ.
Классификация алгоритмов моделирования:
- итерационные;
- событийные.



Преимущества и недостатки:
Последовательная реализация алгоритма простой итерации:
- преимущество: простота реализации;
- недостаток: большое время вычисления, большие вычислительные затраты.

Параллельная реализация алгоритма простой итерации:
- преимущество: простота реализации, ускорение за счет меньшего числа обращений к описанию схемы;
- недостаток: большое время вычисления, большие вычислительные затраты.

Алгоритм ускоренной итерации (Зейделя):
- преимущество:ускорение за счет использования на текущей итерации значений, вычисленных на этой же итерации;
- недостаток: необходимость предварительного ранжирования элементов логической схемы.

Событийный алгоритм моделирования:
- преимущество: ускорение за счет просчета только тех элементов, у которых изменилось значение сигнала хотя бы на одном входе;

недостаток: необходимость выделения оперативной памяти для хранения таблицы текущих событий и таблицы будущих событий.

2. Вероятностное тестирование.

Вероятностное тестирование характеризуется тем, что на входы проверяемого устройства подаются случайные или псевдослучайные последовательности. Одна из возможных схем вероятностного некомпактного тестирования приведена на рис.3.8. Случайные последовательности подаются на входы проверяемого и эталонного устройств, а выходы обоих устройств сравниваются между собой. При вероятностном компактном тестировании на входы проверяемого устройства подаются случайные тестовые наборы, а результаты тестирования сжимаются одним из способов, приведенных в предыдущем параграфе, и сравниваются с эталонным сжатым результатом (рис.3.9).Вероятностное компактное тестирование (ВКТ) выполняется за два или три шага в зависимости от проверяемой схемы - комбинационной или с памятью. На первом шаге для схем с

памятью, носящем название *инициализации,* на них подается длинная последовательность случайных наборов, цель которой – установка схем в исходное состояние. Следующие два шага аналогичны для комбинационных схем и схем с памятью: *накопление результата* и *сравнение со сжатым эталоном.* Сжатый результат даже в случае неисправности проверяемой схемы может не дать точного совпадения с эталоном, а быть лишь достаточно близким к нему. Причиной этого может быть не зафиксированное начальное состояние тестируемых схем с памятью или генераторов случайных наборов, а также вхождение в набор запрещенных входных воздействий. Поэтому при ВКТ проверяемая схема считается исправной, если результат отличается от эталона на величину, не превышающую ξ и называемую *допустимым отклонением* и при кажущейся простоте применения случайных испытательных последовательностей, на практике обычно требуются длительные эксперименты по подбору характера входных воздействий для получения стабильных реакций ОК

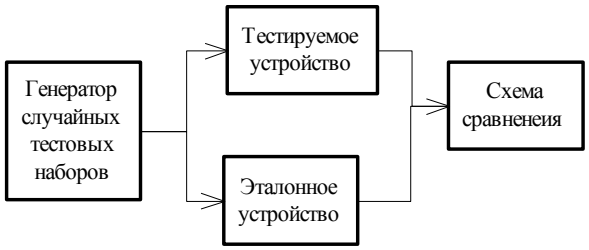


Рис.3.8.