

Задача 1.

При реализации табличного счёта (по умолчанию – счет на возрастание) ячейки ПЗУ содержат адреса следующей ячейки, т.е.:

Адрес ячейки	Содержимое ячейки
000	001
001	010
...	...
111	000

Ответ: 11111101

Задача 2.

Задержка в ОУ равна наибольшему времени выполнения МК.

Ответ: 25

Задача 3.

В данной задаче разрядность ячеек ЗУ является лишней информацией, т.к. дешифрации подлежит только адрес ячеек ЗУ. Для построения ЗУ, имеющего 4096 адресов, из ЗУ, имеющего 1024 адреса, потребуется 4 микросхемы. Соответственно, дополнительный дешифратор должен иметь 4 выхода, следовательно, входов должно быть 2.

Вся емкость пзу 4к, имеется микросхема 1к, значит надо 4 микросхемы, и 16 разрядов на выходе, значит всего 8 микросхем. Дополнительно надо адресовать 4 микросхемы, соотв-нно у дополнительного дешифратора будет 2 входы 4 выхода, который будет выбирать соответствующие блоки микросхем.

Ответ: 2

Задача 4.

Чтобы получить необходимую разрядность, потребуется $32/8=4$ микросхемы (на один адрес). Для увеличения количества адресов в $4096/1024=4$ раза, потребуется 4 микросхемы. Получаем, что необходимо $4*4=16$ микросхем.

4к 32 разряда, 1к 8 разрядов, значит по адресам потребуется в 4 раза больше микросхем, значит по 4 микросхемы на адрес, и по 32 разряда, по 4 на разряд, $4*4 = 16$ корпусов микросхемы

Ответ: 16

Задача 5.

Стек растёт в сторону младших адресов, значит при записи в стек, содержимое регистра указателя стека будет уменьшаться, а при записи – увеличиваться.

Растет в сторону младших адресов, значит адрес уменьшится на 1 соответственно.

Ответ: 0110

Задача 6.

Условие аналогично предыдущей задаче, только тек растёт в сторону старших разрядов, значит при записи счётчик будет увеличиваться, а при записи – уменьшаться.

Аналогично предыдущей. Т.е при чтении стек будет уменьшаться. В предыдущей задаче адрес будет увеличиваться, в этой будет уменьшаться.

Ответ: 0110

Задача 7.

Состав МК:

- Два адреса следующей МК (двухадресный формат)
- Логическое условие (ЛУ)
- Операционная часть (ОЧ)

Для адресации 100 МК (100 – из условия задачи) потребуется 7 ($\lceil \log_2 100 \rceil$) разрядов (7-входовой ДШ способен распознать 128 значений). Количество разрядов для ЛУ считается аналогично – $\lceil \log_2 7 \rceil = 3$. МО закодированы горизонтальным методом, значит каждой МО соответствует свой разряд в ОЧ (10 МО -> 10 разрядов). Складываем все разрядности и получаем разрядность одной МК ($2*7+3+10=27$). Для получения объёма программы умножаем разрядность МК на количество МК в программе: $27*100=2700$.

100 МК всего, значит надо определить разрядность МК, 7 логических условий это 3 разряда, 10 МО – 10 разрядов, значит для 100 МК потребуется 7 разрядов, и соответственно складываем все разряды всех полей и получаем сперва сколько бит в одной МК, увеличиваем на 100 и получаем объем микропрограммы. 2-х адресный формат -> будет два адреса.

Ответ: 2700.

Задача 8.

Полностью аналогичная задача, только в МК содержится один адрес вместо двух, и добавляется поле инверсии ЛУ. Разрядность МК = $7+3+1+10=21$. Объём программы = $21*100=2100$.

Т.е адрес делить только один, 7 разрядов на адрес. Еще одно поле добавится на инверсию. 10 разрядов, складываем общий объем, умножаем на 100, получаем..

Ответ: 2100.

Задача 9.

При смешанном кодировании МО объединяются в группы МО, которые никогда не выполняются вместе. В данной задаче:

a	b	c	d
V1	V2	V3	V6

V5	V4	-	-
----	----	---	---

Получили 4 группы, каждую из которых можно закодировать 1 битом. При последовательном выполнении МК, в МК содержатся только коды МО, разрядность кодов МО равна 4. Программа состоит из 3х МК, значит вес программы равен $3*4=12$.

Определить надо сколько бит. По одному биту на МО, которая горизонтально кодируется, и с использованием дешифратора два в степени n, т.е степень двойки. Горизонтально – МО выполняются одновременно. Вертикально – те, которые можно закодировать через дешифратор, т.е те, которые одновременно не выполняются, разделяем МК по группам, и кодируем, которые вып-ся одновременно горизонтально, и вертикально не выполняются. Должна быть нулевая МО на выходе, т.е когда эти МО не должны выполняться.

Ответ: 12.

Задача 10.

Для адресации МК в памяти микропрограммы потребуется $\lceil \log_2 2048 \rceil = 11$ разрядов. Соответственно, ссылка на любую МК из ПНА должна быть такой же разрядности. Получаем, что ПНА содержит 256 (256 – из условия) 11-разрядных слов. Объём памяти ПНА = $256*11=2816$.

Преобразователь адреса для 256 различных операций – у него должно быть на выходе 8 разрядов. Значит объём памяти 2к - 11 разрядов для адреса. Значит всего в ПЗУ должно быть 256 ячеек, каждая размером 11 разрядов. Каждая ячейка адресует одну операцию, разрядность адреса 11. Умножаем и получаем.

Ответ: 2816.

Задача 11.

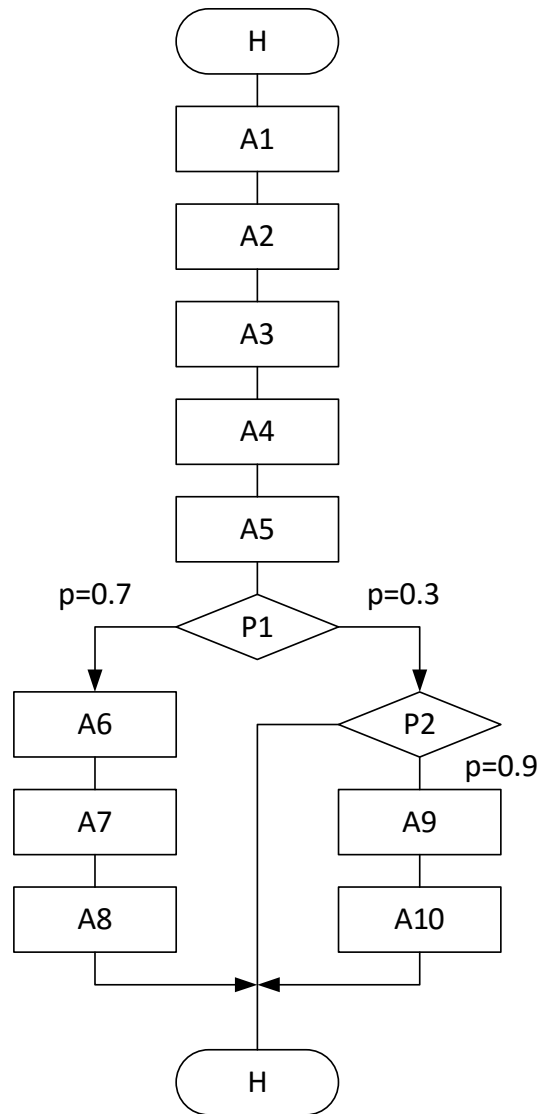
Номинальное быстродействие – величина, обратная средневзвешенному времени выполнения МП, которое в данной задаче равно $(100*10^{-9}*0,1+10*10^{-9}*0,9 = 19*10^{-9})$. Номинальное быстродействие = $1/(19*10^{-9}) = 52.631.578$ операций/сек.

Формула сумма n итое на t итое – формула быстродействия единицы деленное на время, нс = 10^{-9} .

Ответ: 52631578.

Задача 12.

По данным задачи составляем ГСА:



$$N = 5 + (0.7 \cdot 3) + (0.3 \cdot (0 + (0.9 \cdot 2))) = 2.64$$

$$T = t \cdot N = 0.1 \cdot 10^{-6} \cdot 2.64 = 0.264 \text{ мкс}$$

Рисуем алгоритм. Вероятность P2 и 1-P2 это условное ветвление, одна ветвь идет с вероятностью P2 другая с вероятностью 1-P2. Умножаем все ветви с заданной вероятностью, на количество МО, потом всё складываем. И определяем средневзвешенное время выполнения микропрограммы.

Ответ: 0.264.

Задача 13.

Задержка ВУ равна сумме задержек ОУ и УУ.

Указаний нету, значит используется бесконвеейрный способ, сперва работает устройство управления. Потом операционное устройство. Время операционного устройства определяется по максимальному времени. Соответственно складываем задержку на устройстве управления и на ОУ, получаем...

Ответ: 175.

Задача 14.

Время выполнения МП: $T_{МП} = t_{МК} * n_{МК}$

$$T_{МП1} = t_{МК1} * n_{МК1}$$

$$T_{МП2} = 0.6 * t_{МК} * 1.1 * n_{МК}$$

$$T_{МП1} / T_{МП2} = 1 / (0.6 * 1.1) = 1.5$$

Исходное время берем 100% и потом увеличиваем или уменьшаем по условиям задачи, результат в размах.

Ответ: 1.5.

Задача 15.

Время заполнения конвейера = ((количество ступеней конвейера)-1)*(время такта). После заполнения конвейер выполняет по порядку n (в данной задаче n=100) МК. Т.е. общее время работы конвейера (по условиям задачи) = $(10-1)*100*10^{-9} + 100*100*10^{-9} = 10.9$ мкс.

Заполнение конвейера на десятиступенчатом $9*100$ нс, в каждом такте будет выдаваться результат выполнения операции. Т.е умножаем $100*$ время такта + время заполнения конвейера.

Ответ: 10.9

Задача 16.

Нужно посчитать количество повторно использованных функций (повторяющиеся использования тоже считать).

Рисуем схему и смотрим по логическим выражениям.

Ответ: 1.

Задача 17.

(Это пиздец, а не задача)

Сначала нужно посчитать, сколько единиц оборудования займет ВУ без функциональной интеграции:

- Умножалка – 100 ед.
- Складывалка – 40 ед.
- Вычиталка – 80% от 40 = 32 ед.

Теперь делим затраты ВУ с интегрированием (100) на затраты ВУ без интегрирования ($100+40+32 = 172$). $100/172 = 58.13\%$

Исходное 100%, 40% - для сложения, 80% от 40 – 32 – всего 72 процента сокращения.

Ответ: 58.3%

Задача 18-20: читать про способы адресации

18. Неосредственно адресация – в адресной части задается операнд в формате команды. В данном случае – 8. Косвенная регистровая адресация в адресной части задается номер регистра, содержащая адрес ячейки памяти, в которой содержатся исходные данные. В нем записана 8-ка, значит будет рассматриваться регистр 8й. В этом регистре записано значение 9. Будет рассматриваться 9 ячейка памяти, в ней записано.. Будет складываться ...

19. Косвенная регистровая автодекрементная адресация – берется регистр 8й. Берутся данные из этого регистра – 9. До выполнения из них вычитается 1ца. Получается 8. Автодекрементная т.е вычитается 1ца до выполнения. Смотрится ячейка 8я ячейка. В ней код 11. Дальше конъюнктивное умножение младшем разряде 1, остальные нули. Получается 1.

20. Используется автоинкрементная адресация. При ней содержимое регистра увелич после выполнения команды. В 8 регистре записан код 9. Используется 9 ячейка памяти. Результат конъюнктивного умножения 0.