Разные учебные машины

1) Ячейка (в учебных машинах) – минимальная адресуемая часть памяти (конструктивно делится на более мелкие единицы - разряды).

2) Адрес ячейки – это её порядковый номер (при нумерации ячеек от нуля целыми числами).

3) Машинное слово – это содержимое ячейки памяти (при этом машинное слово может представлять как команду, так и число).

4) Бит – это содержимое отдельного разряда (в нем может храниться 0 или 1).

5) Регистр – это ячейка, которая находится не в основной (оперативной) памяти, а *в другом устройстве*, например, в ЦП (центральном процессоре), устройстве ввода-вывода и др. Доступ к регистру, как правило, в несколько раз быстрее, чем к обычной ячейке оперативной памяти.

6) Дробно-адресная ЭВМ – ЭВМ, имеющая адресуемые регистры (т.е. регистры, к которым можно обращаться из программы – для хранения часто используемых величин, выполнения промежуточных вычислений и т.п.).

7) Машинная программа – набор записанных в памяти (не обязательно последовательно) машинных команд, описывающих шаги работы алгоритма.

8) Машинная команда – приказ машине выполнить одну из ее операций.

9) Код операции – это номер (целое число) той операции, которую следует выполнить согласно машинной команде (хранится в отдельном поле машинной команды).

10) Регистры операндов – это внутренние (т.е. неадресуемые) регистры в ЦП, на которые помещаются исходные данные перед выполнением заданной в команде операции.

11) Сумматор – это внутренний (т.е. неадресуемый) регистр в ЦП, на который помещается результат выполнения заданной в команде операции.

12) Как в ЭВМ определяется, что в данный момент находится в ячейке: *данное* или *команда*?
Команды и данные по внешнему виду (в двоичном представлении) не различаются. Когда содержимое ячейки поступает в регистр команд, то оно трактуется как *команда*, а когда содержимое ячейки поступает на регистры операндов, то оно трактуется как *данное*.

13) Как в ЭВМ определяется, какое число – *со знаком* или *без знака* находится в ячейке? Числа со знаком и без знака по внешнему виду (в двоичном представлении) не различаются. Трактовка числа как *знакового* или как *беззнакового* зависит от кода операции конкретной команды.

14) В чем преимущество **УМ-2** по сравнению с **УМ-3**?
УМ-3: формат команд - `КОП A1,A2,A3` действие – `A1:=<A2>o<A3>.`
УМ-2: формат команд - `КОП A1,A2,` действие – `A1:=<A1>o<A2>.`
УМ-3: в рамках одной команды часто требуются одни и те же адреса (например, при x:=x*x или x:=x*y) или не все адреса используются (в командах пересылки, останова, ввода-вывода), что приводит к разрастанию программы. При переходе к **УМ-2** эта проблема частично (при x:=x*x все равно 2 адреса повторяются) решается и => длина команды уменьшается. Но количество команд немного увеличиваются. Однако, в целом, достигается сокращение размеров программы.
Недостаток **УМ-2** по сравнению с УМ-3: менее естественна для человека.

15) Почему (при одинаковой технической реализации) **УМ-1 работает быстрее УМ-2?**
УМ-2: формат команд - `КОП A1,A2`, действие – `A1:=<A1>o<A2>,`
3 обращения к памяти (два – для считывания операндов и одно – для записи результата).
УМ-1: формат команд – `КОП A`, действие – `S:=<S>o<A>,` где `S` сумматор (регистр в ЦП), =>
2 обращения к быстрому регистру (для считывания 1-го операнда и для записи результата, и одно обращение к более медленной оперативной памяти – для считывания 2-го операнда).

16) Почему (при одинаковой технической реализации) **УМ-1 работает быстрее УМ-3?**
УМ-3: формат команд - КОП A1,A2,A3 действие – A1:=<A2>o<A3>,
3 обращения к памяти (два – для считывания операндов и одно – для записи результата).
УМ-1: формат команд – КОП A, действие – S:=<S>o<A>, где S сумматор (регистр в ЦП), =>
2 обращения к быстрому регистру (для считывания 1-го операнда и для записи результата, и одно обращение к более медленной оперативной памяти – для считывания 2-го операнда).

17) Преимущества и недостатки стековых машин (**УМ-С**) по сравнению с другими архитектурами.
Память в них делится условно на 2 части: ОЗУ (оперативная память) и СТЕК.
Преимущества УМ-С. В арифметических командах не надо указывать местоположение операндов (они находятся в двух верхних словах стека) и не надо указывать местоположение результата (т.к. он помещается в стек) => в команде задается только КОП (команда стала короткой), что приводит к компактной программе.
Недостатки УМ-С. Требуется больше команд. Необходимо дополнительно иметь одноадресные команды для обмена с ОЗУ, для переходов. СТЕК – это часть оперативной памяти (медленной) => стековая машина работает медленно (иметь стек на регистрах - дорого). Кроме того, программировать только на стеке - неудобно.

18) Ответить на вопросы по учебным машинам: а) В чём назначение регистра счётчика адреса? б) Может ли программист изменить значение этого регистра, если да, то каким образом?
Во время выполнения текущей команды в регистре счётчика адреса хранится адрес следующей по порядку команды. Программист может изменить значение этого регистра только с помощью команд переходов.

19) В чем преимущества и недостатки дробно-адресной машины (**УМ-Р**)?
Преимущества УМ-Р. В УМ-Р имеются регистры (быстрые ячейки в процессоре), которые можно задействовать для хранения часто используемых в ходе вычислений величин (особенно в цикле). Тем самым *снижается количество обращений к основной (более медленной) памяти* и => *повышается быстродействие* программы.
Недостатки УМ-Р. Для каждой арифметической операции необходимо иметь по 2 варианта команд: регистр-регистр и регистр-память. Это приводит к *усложнению* и *удорожанию* ЦП. Кроме того, память на регистрах – дорогая (поэтому регистров не бывает много).

20) Общая схема выполнения команд в учебных машинах (только для **УМ-3, УМ-2, УМ-1, УМ-С**):

repeat

1. RK := <RA>; чтение очередной команды из ячейки памяти с адресом RA на регистр команд RK

2. RA := RA + 1; увеличение счётчика адреса на единицу, чтобы следующая команда (если текущая команда не является командой перехода) выполнялась из следующей ячейки памяти.

3. Выполнение в АЛУ операции, заданной в коде операции (КОП) (см. ниже подробно про каждую УМ).

until (Err = 1) **or** (КОП = СТОП)

Детализация шага **3** «выполнение операции» (применительно к двуместным арифметическим операциям):

УМ-3: R1 := <A2>; R2 := <A3>; S := R1⊗R2; <A1> := S;

УМ-2: R1 := <A1>; R2 := <A2>; S := R1 ⊗ R2; <A1> := S;

УМ-1: R1 := <A1>; S := S ⊗ R1; (+ доп. команды записи/чтения на сумматор)

УМ-С: R1 := ИЗСТЕКА; R2 := ИЗСТЕКА; S := R1 ⊗ R2; ВСТЕК(S) (+ доп. команды записи/чтения в стек)

Обозначения: RK – регистр команды; RA – регистр счётчика адреса; R1 и R2 – регистры 1-го и 2-го операндов; S – сумматор (регистр результата); Err – регистр ошибки, ⊗ - условное обозначение выполняемой арифметической операции.