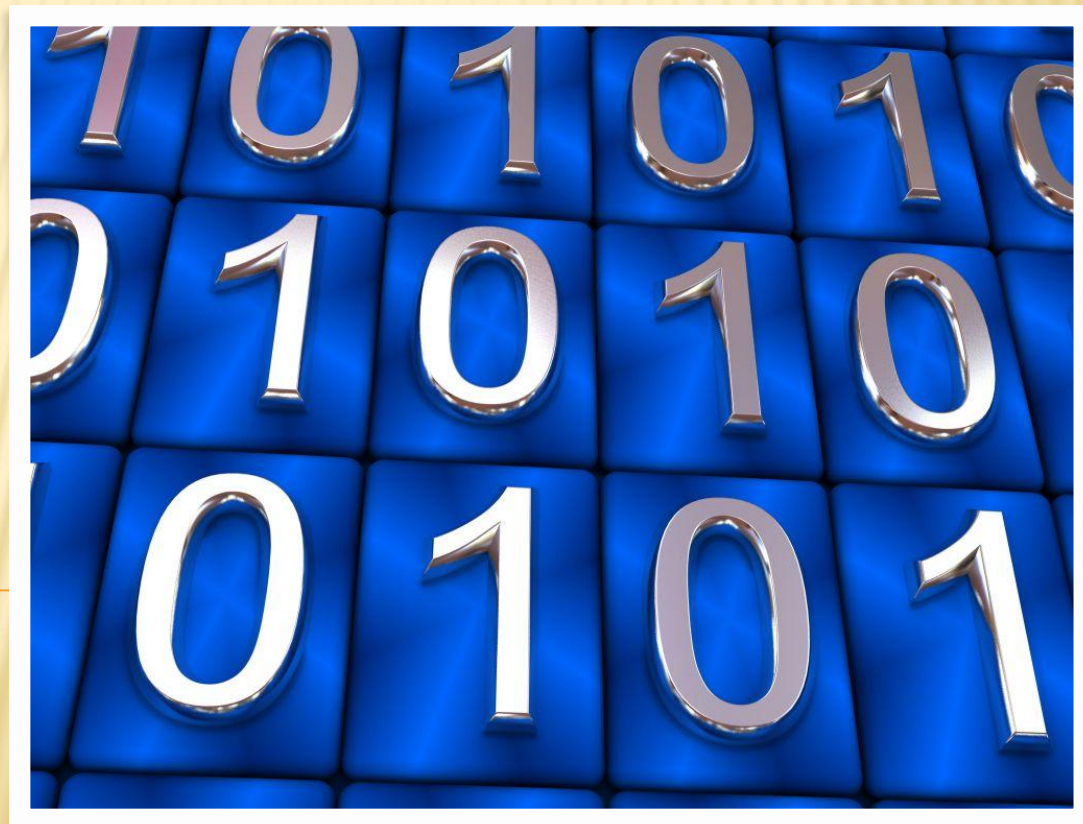


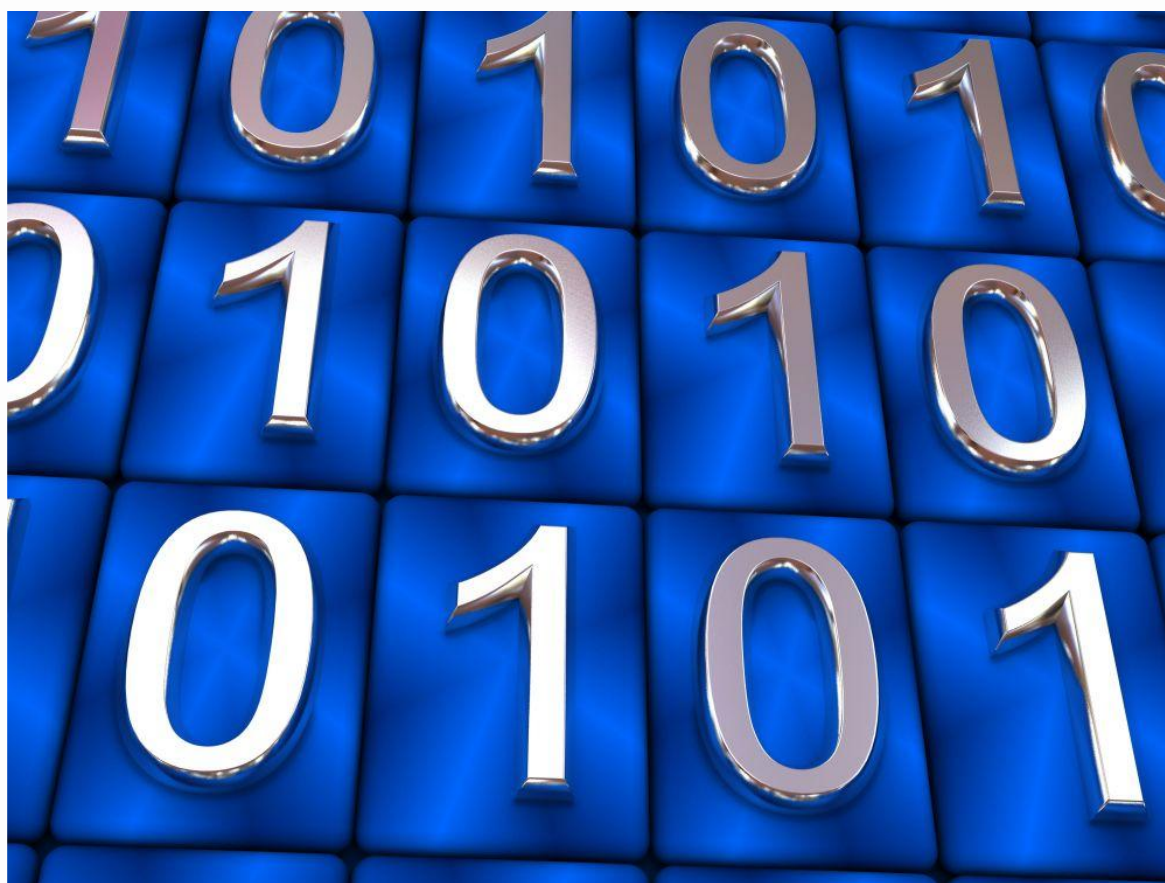
ЯЗЫКИ

ПРОГРАММИРОВАНИЯ



□ Язык программирования — формальная знаковая система, предназначенная для записи компьютерных программ. Язык

проц
лекс
зада



ил,

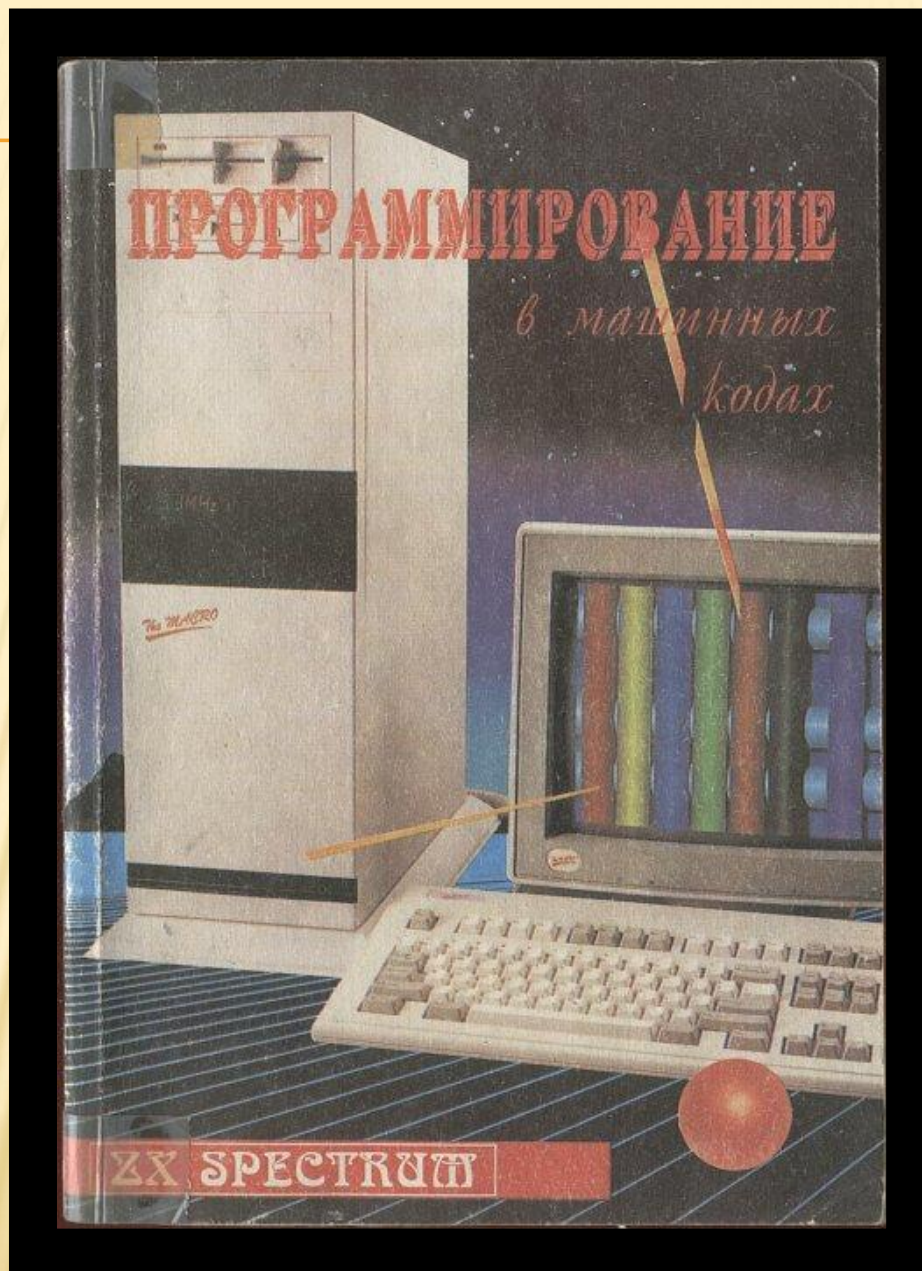
ПЕРВЫЕ УНИВЕРСАЛЬНЫЕ ЯЗЫКИ

Первые программы писались на машинном языке. Программисты обязаны были знать архитектуру машины досконально. Программы были достаточно простыми, что обуславливалось, во-первых, весьма ограниченными возможностями этих машин, и, во-вторых, большой сложностью разработки и, главное, отладки программ непосредственно на машинном языке. Вместе с тем такой способ разработки давал программисту просто невероятную власть над системой. Становилось возможным использование хитроумных алгоритмов и способов организации программ. Например, могла применяться такая возможность, как самомодифицирующийся код. Знание двоичного представления команд позволяло иногда не хранить некоторые данные отдельно, а встраивать их в код как команды.

ПРОГРАММИРОВАНИЕ

*в машинных
кодах*

ZX SPECTRUM



СИСТЕМА КОМАНД ЭВМ-220

Операции над числами (Р, Q, T - порядки чисел X, Y, Z по A ₁ , A ₂ , A ₃)						Логические операции (F, G, S - значения кодов по A ₁ , A ₂ , A ₃)						Операции сдвига						Команды передачи управления с изменением [P]					
Операция	КОП				ω=1	t, мксек	Операция	КОП	Результат	ω=1	t, мксек	Операция	КОП	Результат	ω=1	t, мксек	КОП	Код в РА	Передача управл. в A ₂	Передача управл. в КРА+1	t, мксек		
Сложение	01	21	41	61			Сравнение	15	$C_k = F_k - T_k$			Сдвиг кода по A ₁	54	$C_k = F_k - T_k; S = P - 100$	C=0	24 65	11		PA < A ₁ и ω=1	PA > A ₁ или ω=0			
Вычитание	02	22	42	62	Z < 0	285	Сравнение с ост.	35	аналогично 15, но OCT при C=0	C=0	24	Сдвиг кода по P	74	$C_k = F_k; S = P - 100$	C=0	24 65	31		PA > A ₁ и ω=1	PA < A ₁ или ω=0			
Вычитание модулей	03	23	43	63			Логическое умножение	55	$C_k = F_k \wedge T_k$			Сдвиг мантиссы по A ₁	14	$C_k = F_k; S = A_1^{100} - 100$	C=0	24 65	51		PA < A ₁ и ω=0	PA > A ₁ или ω=1			
Деление	04	24	-	-		-107	Логическое сложение	75	$C_k = F_k \vee T_k$			Сдвиг мантиссы по P	34	$C_k = F_k; S = P - 100$	C=0	24 65	71		PA > A ₁ и ω=0	PA < A ₁ или ω=1			
Умножение	05	25	45	65	Y=100	51	Специальные операции над кодами					Операции засылки					12						
Извлечение кв. корня	44	64	-	-		272	Цикл сложения	07	$C_k = (F_k + T_k) \bmod 2^8$			Выборка из динамики	17	Засылка кода [Z] → Y			32		PA > A ₁	PA < A ₁			
Выбор младших разрядов произведения	47					24	Цикл вычитания	27	$C_k = (F_k - T_k) \bmod 2^8$			Засылка в динамику	37	Засылка кода [Y] → Z			40		PA < зона A ₂ [A ₁]	PA > зона A ₂ [A ₁]			
Операции изменения порядка							Сложение мантисс	13	$C_k = F_k; S = (F_k + T_k) \bmod 2^8$			Засылка кода	00	[A ₁] → A ₂			60		PA > зона A ₂ [A ₁]	PA < зона A ₂ [A ₁]			
Операция	КОП	Результат	ω=1	t, мксек			Вычитание мантисс	33	$C_k = F_k; S = (F_k - T_k) \bmod 2^8$			Засылка кода с КЗУ	20	[KЗУ] → A ₁					Команды передачи управления с занесением в A ₃				
Сложение порядка с A ₁	06	$T = Q + (A_1^{100} - 100)$					Сложение КОПов	53	$C_k = (F_k + T_k) \bmod 2^8$			Анализ окончания раб. АМ	20	A ₁ → 0000 - переход к 30			16	010000 A ₁ 0000	Всегда	—			
Сложение порядка с P	26	$T = Q + (P - 100)$	Y=100	24			Вычитание КОПов	73	$C_k = (F_k - T_k) \bmod 2^8$			Ма (КОП 50)					56		Всегда	—			
Вычитание порядка из A ₁	46	$T = Q - (A_1^{100} - 100)$					Цикл сдвига	67	$C_k = (F_k \bmod 2^8) \bmod 2^8$			Мб (КОП 70)					36		ω=1	ω=0	24		
Вычитание порядка из P	66	$T = Q - (P - 100)$					Операции в М2, выполняемые по сигналам из М1					Режим					76		ω=0	ω=1			
Команда останова							Операции выполнения по командам Ма и Мб					УЧ					A ₁						
КОП	Совмещенные операции						Выборка команды в М2 по адресу из М1	AP2	40-38P 34P 24-13P	РП адреса без адрес в М2	контр команды	б/я	б/я	б/я	б/я	б/я	б/я	б/я	б/я	б/я	б/я	б/я	
77	Вызов (A ₁) и (A ₂) на P ₁ и P ₂ гашение КЗ при продолжении						Обмен кодами М1 ↔ М2	AP1	40-38P 36-25P 24-13P	РП А2 М2	УЧ АН МЗУ	х	х	х	х	х	х	х	х	х	х	х	
Операции изменения [P]						Операции выполняемые по сигналам от линии связи и аналоговой машины					0010												
КОП	Код в РА	Код в A ₃		t, мксек			Информация переданная на РБФ М2 из М1					0020											
52	A ₂	0520000 A ₁ 0000		24			Выборка команды из яч. ки 0022, 0' МЗУ без занесения 0022 на КРА					0500											
72	зона A ₂ [A ₂]	0520000 A ₁ 0000					Выборка команды из яч. ки 0023, 0' МЗУ без занесения 0023 на КРА					0100											
Команды ввода							Автораэрыб от ЛС					0140											
Операция	КОП	A ₁	A ₂	A ₃			Автораэрыб от АМ					0200											
Ввод с останком при несоблюдении КЗ	10	A ₁ МЗУ	КРА	АКΣ			Засылка состояния					0300											
Ввод без ост. при несобл. КЗ	30	A ₁ МЗУ	КРА	АКΣ			Засылка состояния					0040											
Операция изменения регистров приращения							Засылка состояния					0060											
КОП	A ₁	A ₂	код заносимый в A ₃	t, мксек			Засылка состояния					0030											
57	2,1р-РПМ1	32,1р-РПА3		24			Засылка состояния					0034											
54р-РПМЕ	63,4р-РПА2						Засылка состояния					0033											
7р-Пр А3	98,7р-РПА1						Засылка состояния					0031											
8р-Пр А2	2,1р-РПА1						Засылка состояния					0032											
9р-Пр А1	2,1р-РПА1						Засылка состояния					0033											
10р-Пр КРА	2,1р-РПА1						Засылка состояния					0031											
11р-Пр МП	2,1р-РПА1						Засылка состояния					0032											
12р-Пр М5	2,1р-РПА1						Засылка состояния					0033											

1. Емкость промежуточного накопителя вывода ПНВ 1024 слова.
2. После вывода содержимое ПНВ не стирается.
3. При обмене с МБ возможен переход с данного барабана на следующий (4096-4097).
4. Команда останова имеет код только 22.
5. Команда 20 при A₁=0000 используется как команда перехода по признаку работы с аналоговой машиной.
6. В НМЛ МЗ=0 используется только в режиме "Разметка" для записи дефектных зон МЛ.

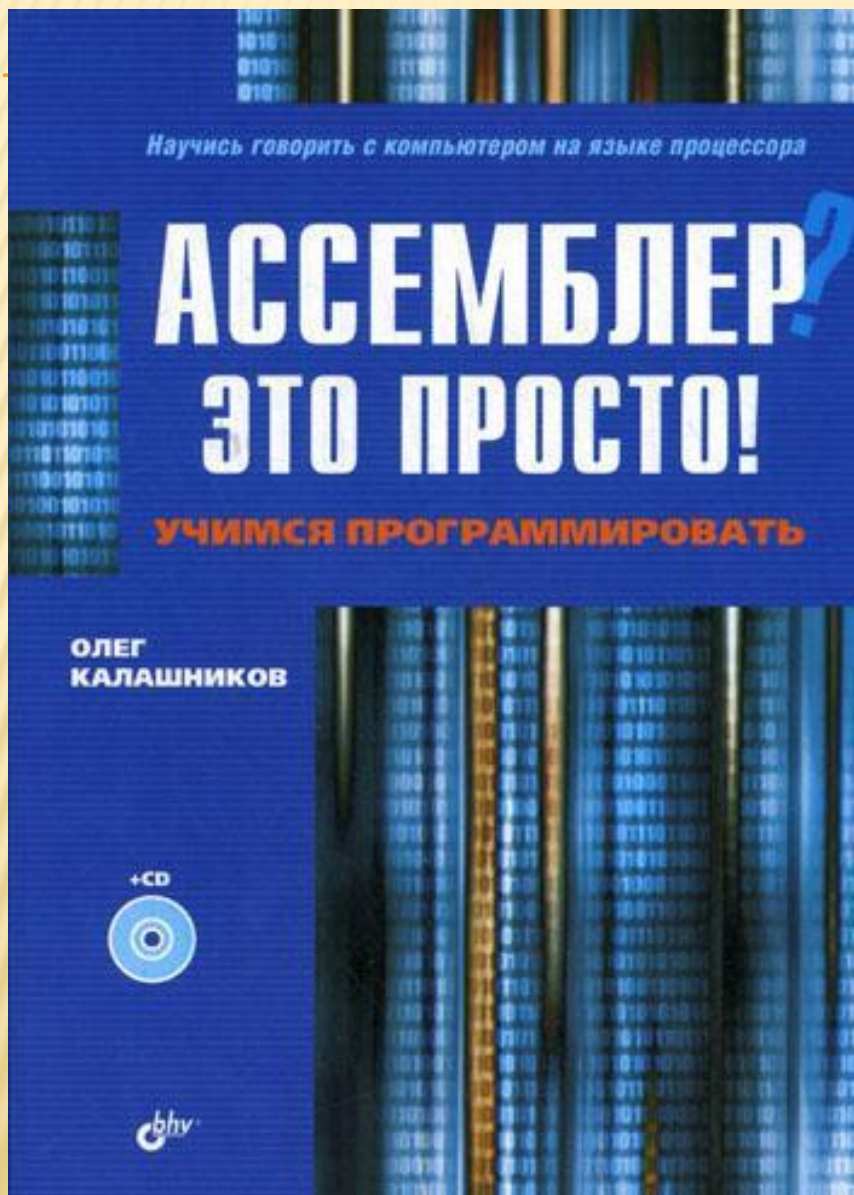
1. При чтении адреса с ПК 1-й РБФ-блокировка АВ ост. при несоблюдении КЗ.
- 1-й 13р РБФ-блокировка записи в МЗУ последующего массива 37-38р РБФ(РПА1) → 13-14р СМ А.

2. При A₁=0 блокируется запись в МЗУ последующего массива.
- Обращение к МЗУ по команде НРП производится по старым значениям НРП МЗУ. РП изменяется только по командам НРП, АРЗБ.

- * Операции АР1, АР2, АР3, АР4, АР5 не изменяют состояния машины.

Примечание: Знак, X означает, что состояние разряда безразлично

Отсутствие знак, означает, что в разряде может быть 0 или 1; A₁-адрес исполнительный; [A₁]-содержимое ячейки с адресом A₁



Первым значительным шагом представляется переход к языку ассемблера. Программисту не надо было больше вникать в способы кодирования команд на аппаратном уровне. Появилась также возможность использования макросов и меток, что также упрощало создание, модификацию и отладку программ.

АССЕМБЛЕР

Вместе с тем, переход к новому языку таил в себе и некоторые отрицательные стороны. Возможности программистов сильно сократились. Кроме того, здесь впервые в истории развития программирования появились два представления программы: в исходных текстах и в откомпилированном виде. К концу ассемблерной эры возможность автоматической трансляции в обе стороны была утеряна. В связи с этим было разработано большое количество специальных программ-дизассемблеров, осуществляющих обратное преобразования, однако в большинстве случаев они с трудом могут разделить код и данные.

Спасибо за внимание!)