

10 апр. 20 г.

Лекция на тему «Язык Техно IL».

Изучить материал, написать конспект в письменном виде в тетрадь. Фото конспекта выслать на почту [nazarovaolga.v@mail.ru](mailto:nazarovaolga.v@mail.ru) до 10 апреля 2020 г.

### Язык Техно IL

Программа на языке Техно IL— последовательность инструкций. Каждая инструкция начинается с новой строки, содержит оператор с модификатором, в случае некоторых операторов приводится один или более операндов. Компилятор не чувствителен к регистру. Инструкции ADD 10 15 и Add 10 15 равнозначны.

В данном языке вводится понятие аккумулятора, под которым следует понимать хранилище текущего результата.

В качестве операндов могут выступать:

1. переменные;
2. константы (см. язык Техно ST);
3. имя метки;
4. имя функции.

Под модификатором следует понимать литеры N, C, X, которые приписываются справа к имени некоторых операторов. Модификатор N обозначает логическое отрицание операнда, C обозначает, что инструкция выполняется, если результат предыдущей операции сравнения истинен, модификатор X указывает на то, что инструкция выполняется, если аккумулятор содержит значение true.

Пример использования модификатора:

OR b//аккумулятор ИЛИ b

ORN b// аккумулятор ИЛИ НЕ b

Операторы, которые предполагают наличие одного или двух операндов (арифметические, логические операции) допускают опустить первый операнд, тогда в качестве первого операнда выступает значение, хранящееся в аккумуляторе. Случай, когда используются оба операнда называется двухадресным режимом, случай, когда первый операнд заменяется аккумулятором— одноадресным режимом.

Пример двухадресного режима:

ADD 10 15//сложение 10 и 15

Пример того же сложения при одноадресном режиме:

LD 10//присвоение аккумулятору числа 10

ADD 15//сложение 10 и 15

Рассмотрим операторы языка Техно IL. Операторы для обмена с аккумулятором приведены в таблице 2.11.

Таблица 2.11

Операторы для обмена с аккумулятором

Обозначение оператора	Синтаксис оператора	Допустимые модификаторы	Действие
LD	LD {операнд}	N	присваивает аккумулятору значение операнда
ST	ST {операнд}	N	присваивает операнду значение аккумулятора

Пример обмена данными с аккумулятором:

LD X//присвоение аккумулятору операнда X

MULT K//Умножение X на K

ST X//присвоение операнду X значения, хранящегося в аккумуляторе.

Логические операторы приведены в таблице 2.12

Таблица 2.12

Логические операторы

Обозначение оператора	Синтаксис оператора	Допустимые модификаторы	Действие
S	S {операнд}	—	присваивает операнду значение

			true
R	R {операнд}	—	присваивает операнду значение false
AND	AND {операнд 1} {операнд 2}	N	логическое «И»
OR	OR {операнд 1} {операнд 2}	N	логическое «ИЛИ»
XOR	XOR {операнд 1} {операнд 2}	N	Логическое «исключающее ИЛИ»

Пример логических операций:

S X//присвоение операнду X значения true

R Y//присвоение операнду Y значения false

AND X Y//X «И» Y

Арифметические операторы приведены в таблице 2.13.

Таблица 2.13

#### Арифметические операторы

Обозначение оператора	Синтаксис оператора	Допустимые модификаторы	Действие
ADD	ADD {операнд 1} {операнд 2}	—	сложение операндов
SUB	SUB {операнд 1} {операнд 2}	—	вычитание операнда 2 из операнда 1

MUL	MUL {операнд 1} {операнд 2}	—	произведение операндов
DIV	DIV {операнд 1} {операнд 2}	—	Деление операнда 1 на операнд 2

Пример выполнения арифметических операций:

MUL X K

ADD C

Операторы сравнения приведены в таблице 2.14.

Таблица 2.14

#### Операторы сравнения

Обозначение оператора	Синтаксис	Допустимые модификаторы	Действие
GT	GT {операнд 1} {операнд 2}	—	возвращает true, если операнд 1 больше операнда 2
GE	GE {операнд 1} {операнд 2}	—	возвращает true, если операнд 1 не меньше операнда 2
EQ	EQ {операнд 1} {операнд 2}	—	возвращает true, если операнд 1 равен операнду 2
NE	NE {операнд 1} {операнд 2}	—	возвращает true, если операнд 1 не равен операнду 2
LE	LE {операнд 1} {операнд 2}	—	возвращает true, если операнд 1 не больше операнда 2

LT	LT {операнд 1} {операнд 2}	—	возвращает true, если операнд 1 меньше операнда 2
----	-------------------------------	---	---------------------------------------------------

Операторы перехода и вызова функций приведены в таблице 2.15.

Таблица 2.15

#### Операторы перехода и вызова

Обозначение оператора	Синтаксис	Допустимые модификаторы	Действие
JMP	JMP {метка}	C, X	переход к строке с указанной меткой
CAL	CAL {имя функции (параметры)}	C, X	Вызов функции
RET	RET	C, X	Выход из программы

Возможны следующие модификации операторов перехода и вызова:

JMPC— условный переход, выполняемый если результат предыдущей операции сравнения истинен;

JMPX— условный переход, выполняемый если аккумулятор содержит значение true;

CALC— условный вызов функции, выполняемый если результат предыдущего сравнения истинен;

CALX— условный вызов функции, выполняемый если аккумулятор содержит значение true

RETC— условный выход из программы, выполняемый если результат предыдущей операции сравнения истинен;

RETX— условный выход из программы, выполняемый, когда аккумулятор содержит значение true.

Пример выполнения сравнения и перехода:

```
EQ Z 0//проверка на равенство Z нулю
```

```
JMPC label//при равенстве нулю переход к строке с меткой label
```

```
...
```

```
label: LD 1e-9//присвоение аргументу Z значения  $10^{-9}$ 
```

```
ST Z
```

Рассмотрим реализацию примера деления, приведенного на языке Техно SFC, с использованием языка Техно IL. Поскольку существует оператор S в языке Техно IL, глобальную переменную переименуем с S на Sum0. Тогда программа на языке Техно IL примет вид:

```
PROGRAM
```

```
VAR_INPUT Y : REAL; END_VAR
```

```
VAR_OUTPUT Z : REAL; END_VAR
```

```
VAR_OUTPUT Sum : REAL; END_VAR
```

```
VAR_INPUT X0 : REAL; END_VAR
```

```
VAR_INPUT X1 : REAL; END_VAR
```

```
VAR Шаг1 : REAL := 0; END_VAR
```

```
ADD X0 X1//Сложение аргументов
```

```
ST Шаг1//присвоение результата сложения переменной Шаг1
```

```
NE Y 0//Проверка на равенство нулю
```

```
JMPC Деление_на_Y//Переход к делению на Z, если Z не равен нулю
```

```
//Деление на  $1e-9$ 
```

```
DIV Шар1 1e-9
ST Z//присвоение аргументу Z вычисленного отношения
JMP Сложение//Переход к сложению
Деление_на_Y: DIV Шар1 Y//деление на Z
ST Z//присвоение аргументу Z вычисленного отношения
Сложение: ADD Z Sum0//Получение суммы всех результатов
отношения
ST Sum0//присвоение суммы всех результатов сложения аргументу
Sumи глобальной переменной S и глобальной переменной S
ST Sum

END_PROGRAM
```