

# Операции сдвига и циклического сдвига

# 20

## Обзор главы

В разделе	Вы найдете	на стр.
20.1	Операции сдвига	20–2
20.2	Операции циклического сдвига	20–6

20.1. Операции сдвига

**Описание**

С помощью операций сдвига Вы можете побитно сдвинуть налево или направо содержимое младшего слова АККУ 1 или содержимое всего аккумулятора. Сдвиг на  $n$  битов влево умножает содержимое аккумулятора на  $2^n$ , сдвиг на  $n$  битов вправо делит содержимое аккумулятора на  $2^n$ . Таким образом, если, например, Вы сдвигаете двоичный эквивалент десятичного числа 3 на 3 бита влево, то в аккумуляторе получается двоичный эквивалент десятичного числа 24. Если Вы сдвигаете двоичный эквивалент десятичного числа 16 на 2 бита вправо, то в аккумуляторе получается двоичный эквивалент десятичного числа 4.

Число, следующее за операцией сдвига или записанное в младшем байте младшего слова АККУ 2, указывает, на сколько битов должен быть произведен сдвиг. Разряды, освобождаемые операцией сдвига, заполняются нулями или сигнальным состоянием знакового бита (“0” - для положительных чисел, “1” - для отрицательных). Последний сдвинутый бит загружается в бит A1 слова состояния (см. 9.1). Биты A0 и OV слова состояния сбрасываются в “0”. Вы можете анализировать бит A1 с помощью операций перехода.

Операции сдвига абсолютны, т.е. их выполнение не зависит ни от каких определенных условий. Они не влияют на результат логической операции.

**Операции сдвига: числа без знака**

Следующие операции сдвигают побитно влево или вправо содержимое младшего слова АККУ 1:

- SLW: Сдвинуть слово (16 бит) влево
- SRW: Сдвинуть слово (16 бит) вправо

Следующие операции сдвигают все содержимое АККУ 1 побитно влево или вправо:

- SLD: Сдвинуть двойное слово (32 бита) влево
- SRD: Сдвинуть двойное слово (32 бита) вправо

Во всех случаях освободившиеся битовые разряды заполняются нулями.

**Сдвинуть влево слово (16 бит):**

Следующий пример программы и рис. 20–1 показывают, как работает операция SLW. В таблице 20–1 Вы найдете обзор всех операций сдвига. **SLW**

AWL	Объяснение
L MW10	Загрузить содержимое меркерного слова MW10 в младшее слово АККУ 1.
SLW 6	Сдвинуть биты в младшем слове АККУ 1 на 6 разрядов влево.
T MW20	Передать содержимое младшего слова АККУ 1 в меркерное слово MW20.

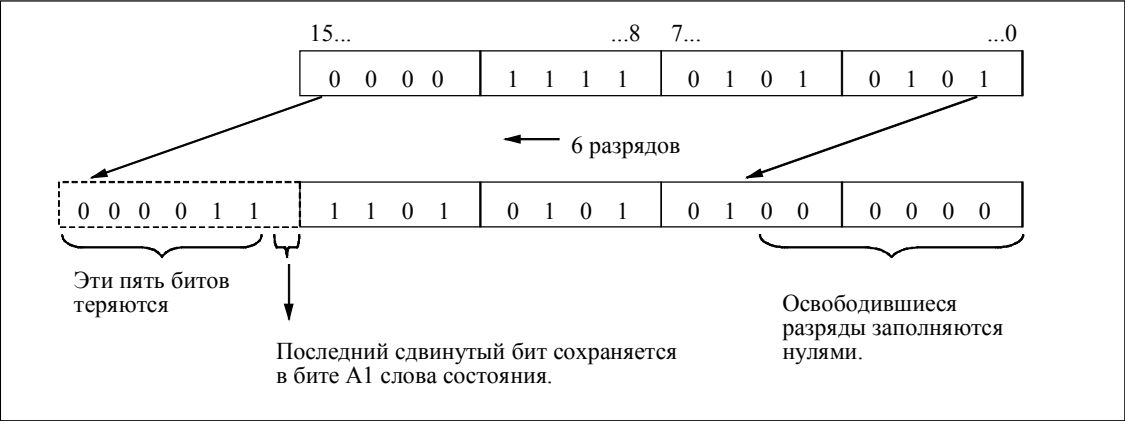


Рис. 20-1. Сдвиг битов в младшем слове АККУ 1 на шесть битов влево

**Сдвинуть вправо  
двойное слово  
(32 бита): SRD**

Следующий пример программы и рис. 20–2 показывают, работает операция SRD. В таблице 20–1 Вы найдете обзор всех операций сдвига.

AWL	Объяснение
L +3	Загрузить значение +3 в АККУ 1.
L MD10	Загрузить содержимое двойного меркерного слова MD10 в АККУ 1. Старое содержимое АККУ 1 (+3) сдвигается в АККУ 2.
SRD	Сдвинуть биты в АККУ 1 на три бита вправо.
T MD20	Передать содержимое АККУ 1 в двойное меркерное слово MD20.

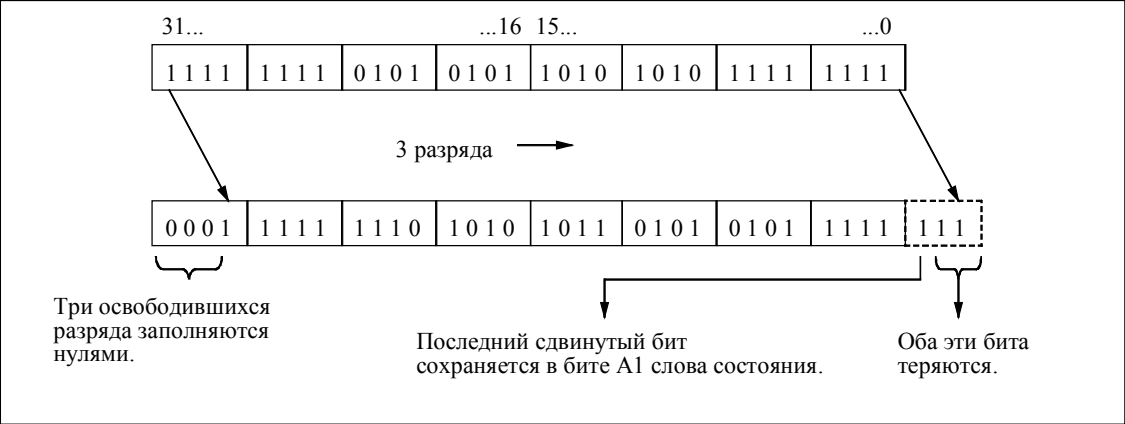


Рис. 20-2. Сдвиг битов АККУ 1 на три разряда вправо

**Операции сдвига:  
числа со знаком**

Операция SSI - Сдвинуть целое число со знаком (16 бит) сдвигает побитно вправо содержимое младшего слова АККУ 1 включая знак.

Операция SSD - Сдвинуть целое число со знаком (32 бита) сдвигает побитно вправо все содержимое АККУ 1 включая знак.

Знаковый бит копируется в освободившиеся битовые разряды.

**Сдвинуть целое  
число со знаком  
(16 бит): SSI**

Следующий пример программы и рис. 20–3 показывают, как работает операция SSI. В таблице 20–1 Вы найдете обзор всех операций сдвига.

AWL	Объяснение
L MW10	Загрузить содержимое меркерного слова MW10 в младшее слово АККУ 1.
SSI 4	Сдвинуть биты в младшем слове АККУ 1 включая знак на четыре разряда вправо.
T MW20	Передать содержимое младшего слова АККУ 1 в меркерное слово MW20.

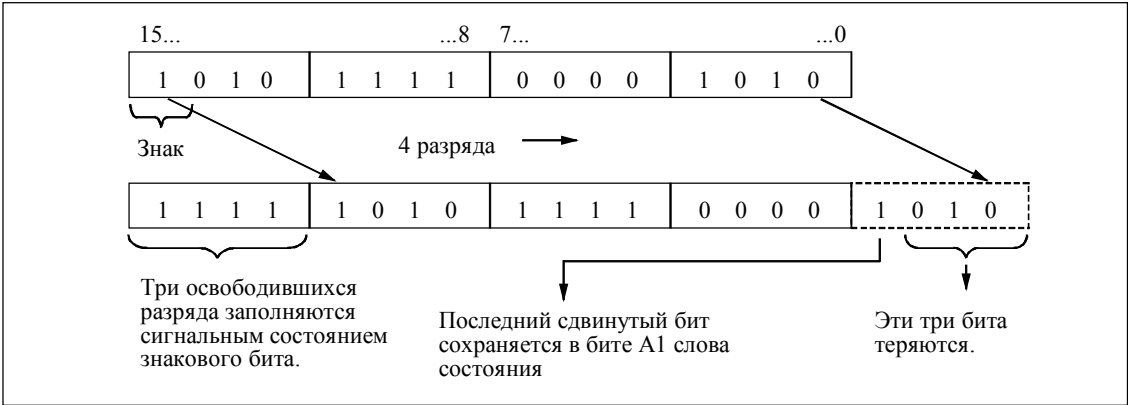


Рис. 20-3. Сдвиг битов младшего слова АККУ 1 включая знак на четыре разряда вправо

Таблица 20–1. Обзор операций сдвига

Операция	Задействованная область	Направление	Указание количества разрядов, на которые должен быть произведен сдвиг	Освобожденные разряды заполняются...	Диапазон сдвига <sup>1)</sup>
SLW n	Младшее слово AKKU 1	влево	в команде	0	n=0 до 15
SLW	Младшее слово AKKU 1	влево	в младшем байте младшего слова AKKU 2	0	0 до 255 <sup>2)</sup>
SLD n	AKKU 1	влево	в команде	0	n=0 до 32
SLD	AKKU 1	влево	в младшем байте младшего слова AKKU 2	0	0 до 255 <sup>3)</sup>
SRW n	Младшее слово AKKU 1	вправо	в команде	0	n=0 до 15
SRW	Младшее слово AKKU 1	вправо	в младшем байте младшего слова AKKU 2	0	0 до 255 <sup>2)</sup>
SRD n	AKKU 1	вправо	в команде	0	n=0 до 32
SRD	AKKU 1	вправо	в младшем байте младшего слова AKKU 2	0	0 до 255 <sup>3)</sup>
SSI n	Младшее слово AKKU 1	вправо	в команде	знаковым битом	n=0 до 15
SSI	Младшее слово AKKU 1	вправо	в младшем байте младшего слова AKKU 2	знаковым битом	0 до 255 <sup>4)</sup>
SSD n	AKKU 1	вправо	в команде	знаковым битом	n=0 до 32
SSD	AKKU 1	вправо	в младшем байте младшего слова AKKU 2	знаковым битом	0 до 255 <sup>5)</sup>

- 1) Если количество разрядов, на которые необходимо произвести сдвиг, равно 0, то операция не выполняется.
- 2) При значении, большем 16, результат функции сдвига равен W#16#0000 и A1 = 0.
- 3) При значении, большем 32, результат функции сдвига равен DW#16#0000\_0000 и A1 = 0.
- 4) При значении, большем 15, результат функции сдвига равен в зависимости от знакового бита (0 или 1) W#16#0000 и A1 = 0 или W#16#FFFF и A1 = 1.
- 5) При значении, большем 31, результат функции сдвига равен в зависимости от знака числа битов, подлежащих сдвигу, DW#16#0000\_0000 (A1 = 0) или DW#16#FFFF\_FFFF (A1 = 1).

## 20.2. Операции циклического сдвига

### Описание

С помощью операций циклического сдвига Вы можете побитно циклически сдвигать все содержимое АККУ 1 направо или налево. Операции циклического сдвига выполняют функции, подобные описанным в разделе 20.1 функциям сдвига. Однако, освободившиеся разряды заполняются сигнальными состояниями битов, выдвигаемых из аккумулятора.

Число, следующее за операцией циклического сдвига, или значение в младшем байте младшего слова АККУ 2 указывает, на сколько разрядов должен быть произведен циклический сдвиг.

В зависимости от операции циклический сдвиг выполняется через бит A1 в слове состояния (см. гл. 9.4). Бит A0 в слове состояния сбрасывается в “0”.

В Вашем распоряжении имеются следующие операции циклического сдвига:

- RLD: Сдвинуть циклически налево двойное слово (32 бита).
- RRD: Сдвинуть циклически направо двойное слово (32 бита).
- RLDA: Сдвинуть циклически налево АККУ 1 через индикатор A1 (32 бита).
- RRDA: Сдвинуть циклически направо АККУ 1 через индикатор A1 (32 бита).

Если число битов, на которое нужно произвести сдвиг, равно 0, то операция не выполняется.

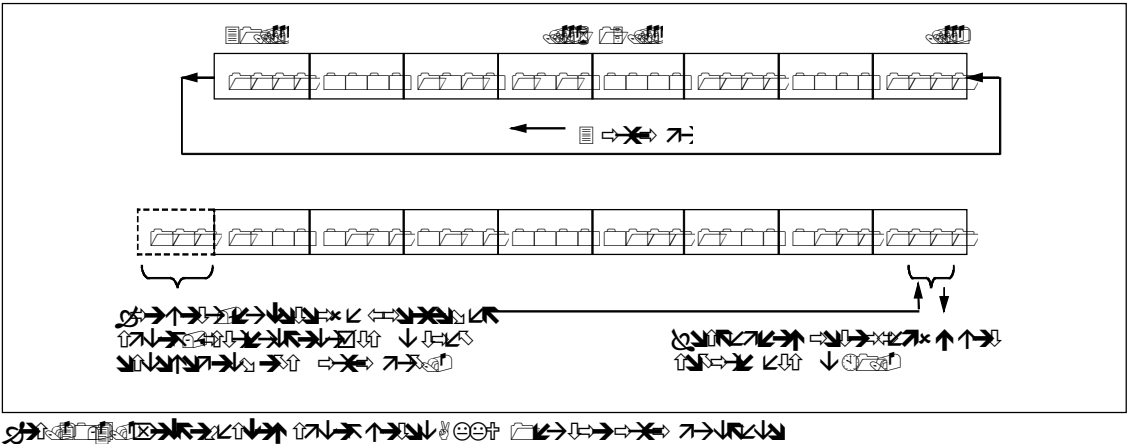
Таблица 20–2. Обзор операций циклического сдвига

Операция	Сдвигать через A1?	Направление	Указание, на сколько разрядов нужно произвести сдвиг	Диапазон сдвига
RLD n	нет	влево	в команде	n=0 до 32
RLD	нет	влево	в младшем байте младшего слова АККУ 2	0 до 255
RRD n	нет	вправо	в команде	n=0 до 32
RRD	нет	вправо	в младшем байте младшего слова АККУ 2	0 до 255
RLDA	да	влево	-	1 (фиксировано)
RRDA	да	вправо	-	1 (фиксировано)

Сдвинуть  
циклически влево  
двойное слово  
(32 бита): RLD

В таблице 20–2 Вы найдете обзор всех операций циклического сдвига.  
Следующий пример программы и рис. 20–4 показывают, как работает операция RLD.

AWL	Объяснение
L MD10	Загрузить содержимое двойного меркерного слова MD10 в АККУ 1.
RLD 3	Сдвинуть циклически биты в АККУ 1 на три разряда влево.
T MD20	Передать содержимое АККУ 1 в двойное меркерное слово MD20.



Сдвинуть  
циклически вправо  
двойное слово  
(32 бита): RRD

Следующий пример программы и рис. 20–5 показывают, как работает операция RRD.

AWL	Объяснение
L +3	Загрузить значение +3 в АККУ 1.
L MD10	Загрузить содержимое двойного меркерного слова MD10 в АККУ 1. Старое содержимое АККУ 1 (+3) сдвигается в АККУ 2.
RRD	Сдвинуть циклически биты в АККУ 1 на три разряда вправо.
T MD20	Передать содержимое АККУ 1 в двойное меркерное слово MD20.

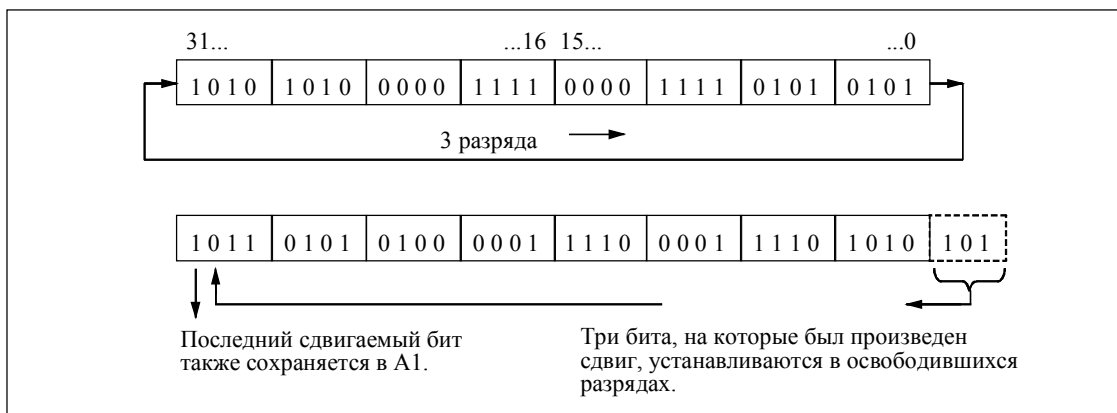


Рис. 20-5. Циклический сдвиг битов АККУ 1 на три разряда вправо

**Циклический сдвиг АККУ 1 влево через индикатор A1** Следующий пример программы и рис. 20–6 показывают, как работает операция RLDA.

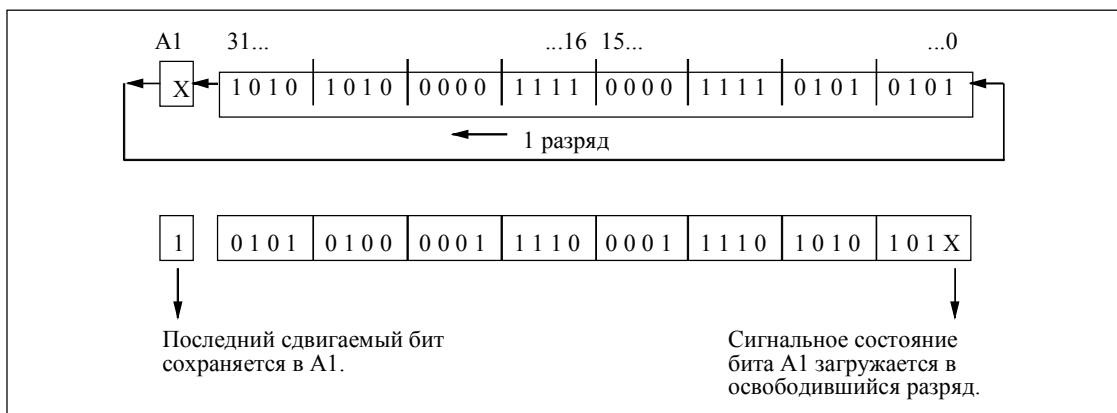


Рис. 20-6. Циклический сдвиг АККУ 1 через бит A1 слова состояния на 1 бит влево

**Циклический сдвиг АККУ 1 вправо через индикатор A1** Операция RRDA работает аналогично операции RLDA; единственное отличие состоит в том, что сдвиг происходит в другом направлении.