

16-разрядная однокристалльная микро-ЭВМ без ПЗУ с RISC-архитектурой 1881BE2T

Микросхема 1881BE2T представляет собой 16-разрядную однокристалльную микро-ЭВМ без ПЗУ с RISC-архитектурой.

Разрабатываемая микросхема предназначена для использования в высокопроизводительных системах обработки информации и устройствах управления, обладает стойкостью к воздействию нейтронного и γ -излучения, ионизирующего излучения космического пространства.

Микросхема изготавливается в 108-выводном металлокерамическом корпусе 4226.108-2.

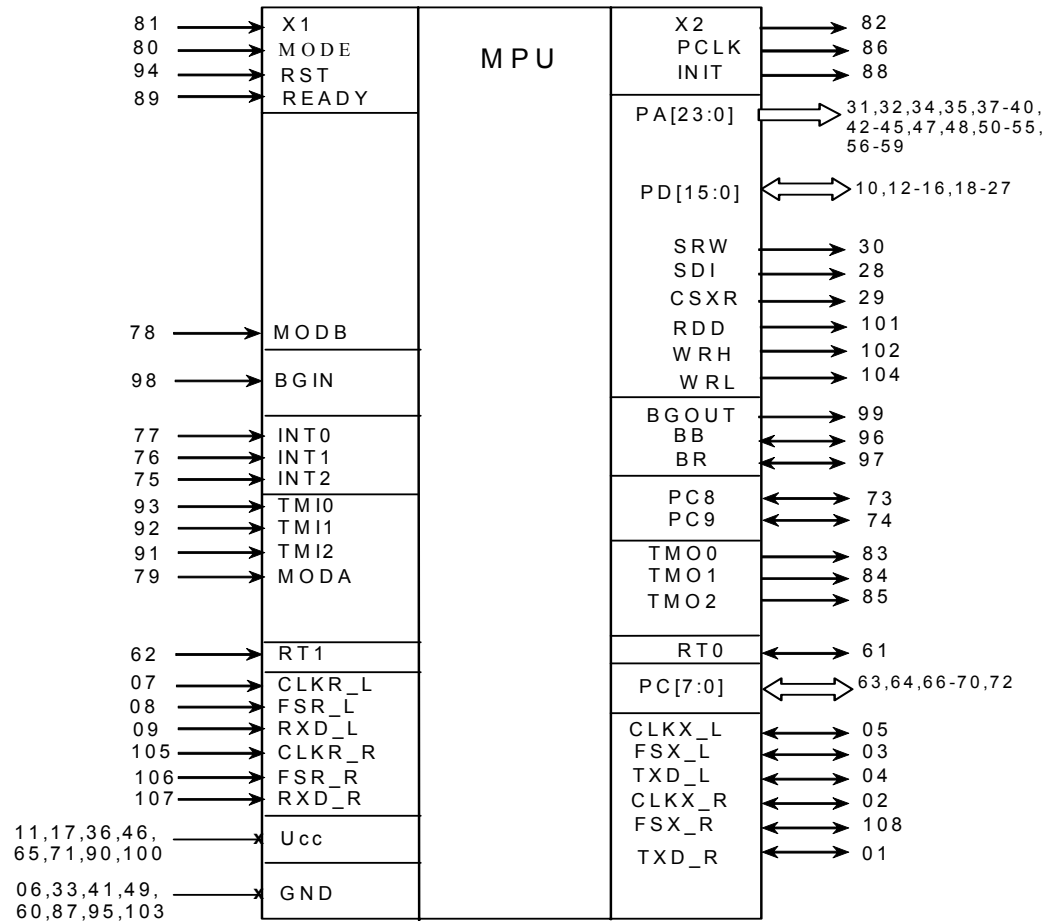


Рисунок 1 – Условное графическое обозначение

Таблица 1 – Назначение выводов

Техническая спецификация 1881 BE2T

Номер вывода	Обозначение	Назначение
01	TXD_R	Вход/выход данных передатчика SSI_R
02	CLKX_R	Вход/выход синхронизации передатчика SSI_R
03	FSX_L	Вход/выход строба передачи SSI_L
04	TXD_L	Вход/выход данных передатчика SSI_L
05	CLKX_L	Вход/выход синхронизации передатчика SSI_L
06	GND	Общий вывод
07	CLKR_L	Вход синхронизации приемника SSI_L
08	FSR_L	Вход строба приема SSI_L
09	RXD_L	Вход данных приемника SSI_L
10	PD15	Вход/выход разряда 15 порта D
11	U _{CC}	Вывод питания от источника напряжения
12	PD14	Вход/выход разряда 14 порта D
13	PD13	Вход/выход разряда 13 порта D
14	PD12	Вход/выход разряда 12 порта D
15	PD11	Вход/выход разряда 11 порта D
16	PD10	Вход/выход разряда 10 порта D
17	U _{CC}	Вывод питания от источника напряжения
18	PD9	Вход/выход разряда 9 порта D
19	PD8	Вход/выход разряда 8 порта D
20	PD7	Вход/выход разряда 7 порта D
21	PD6	Вход/выход разряда 6 порта D
22	PD5	Вход/выход разряда 5 порта D
23	PD4	Вход/выход разряда 4 порта D

Техническая спецификация 1881 ВЕ2Т

Продолжение таблицы 1

Номер вывода	Обозначение	Назначение
24	PD3	Вход/выход разряда 3 порта D
25	PD2	Вход/выход разряда 2 порта D
26	PD1	Вход/выход разряда 1 порта D
27	PD0	Вход/выход разряда 0 порта D
28	SDI	Выход признака команды/данных
29	CSXR	Выход выбора внешнего устройства
30	SRW	Выход признака чтения/записи по порту D
31	PA0	Выход разряда 0 порта A
32	PA1	Выход разряда 1 порта A
33	GND	Общий вывод
34	PA2	Выход разряда 2 порта A
35	PA3	Выход разряда 3 порта A
36	UCC	Вывод питания от источника напряжения
37	PA4	Выход разряда 4 порта A
38	PA5	Выход разряда 5 порта A
39	PA6	Выход разряда 6 порта A
40	PA7	Выход разряда 7 порта A
41	GND	Общий вывод
42	PA8	Выход разряда 8 порта A
43	PA9	Выход разряда 9 порта A
44	PA10	Выход разряда 10 порта A
45	PA11	Выход разряда 11 порта A
46	Ucc	Вывод питания от источника напряжения

Продолжение таблицы 1

Номер вывода	Обозначение	Назначение
47	РА12	Выход разряда 12 порта А
48	РА13	Выход разряда 13 порта А
49	GND	Общий вывод
50	РА14	Выход разряда 14 порта А
51	РА15	Выход разряда 15 порта А
52	РА16	Выход разряда 16 порта А
53	РА17	Выход разряда 17 порта А
54	РА18	Выход разряда 18 порта А
55	РА19	Выход разряда 19 порта А
56	РА20	Выход разряда 20 порта А
57	РА21	Выход разряда 21 порта А
58	РА22	Выход разряда 22 порта А
59	РА23	Выход разряда 23 порта А
60	GND	Общий вывод
61	RT0	Вход/выход передатчика SCI
62	RT1	Вход приемника SCI
63	PC0	Вход/выход разряда 0 порта С
64	PC1	Вход/выход разряда 1 порта С
65	U _{CC}	Вывод питания от источника напряжения
66	PC2	Вход/выход разряда 2 порта С
67	PC3	Вход/выход разряда 3 порта С
68	PC4	Вход/выход разряда 4 порта С

Техническая спецификация 1881 ВЕ2Т

Продолжение таблицы 1

Номер вывода	Обозначение	Назначение
69	PC5	Вход/выход разряда 5 порта С
70	PC6	Вход/выход разряда 6 порта С
71	U _{CC}	Вывод питания от источника напряжения
72	PC7	Вход/выход разряда 7 порта С
73	PC8	Вход/выход I ² C-интерфейса
74	PC9	Вход/выход I ² C-интерфейса
75	INT2	Вход запроса прерывания
76	INT1	Вход запроса прерывания
77	INT0	Вход запроса прерывания
78	MODB	Вход режима работы
79	MODA	Вход режима работы
80	MODE	Вход режима работы
81	X1	Вход подключения внешнего кварцевого резонатора
82	X2	Выход подключения внешнего кварцевого резонатора
83	TMO0	Выход таймера 0
84	TMO1	Выход таймера 1
85	TMO2	Выход таймера 2
86	PCLK	Выход синхросигнала
87	GND	Общий вывод
88	INIT	Выход инициализации
89	READY	Вход готовности данных

Техническая спецификация 1881 ВЕ2Т

Продолжение таблицы 1

90	U _{CC}	Вывод питания от источника напряжения
91	TMI2	Вход управления таймером 2
92	TMI1	Вход управления таймером 1
93	TMI0	Вход управления таймером 0
94	RST	Вход сброса
95	GND	Общий вывод
96	BB	Вход/выход захвата внешней шины
97	BR	Вход/выход запроса внешней шины
98	BGIN	Вход разрешения на захват шины
99	BGOUT	Выход разрешения на захват шины
100	U _{CC}	Вывод питания от источника напряжения
101	RDD	Выход строба чтения по порту D
102	WRH	Выход строба записи по порту D
103	GND	Общий вывод
104	WRL	Выход строба записи по порту D
105	CLKR_R	Вход синхронизации приемника SSI_R
106	FSR_R	Вход строба приема SSI_R
107	RXD_R	Вход данных приемника SSI_R
108	FSX_R	Вход/выход строба передачи SSI_R

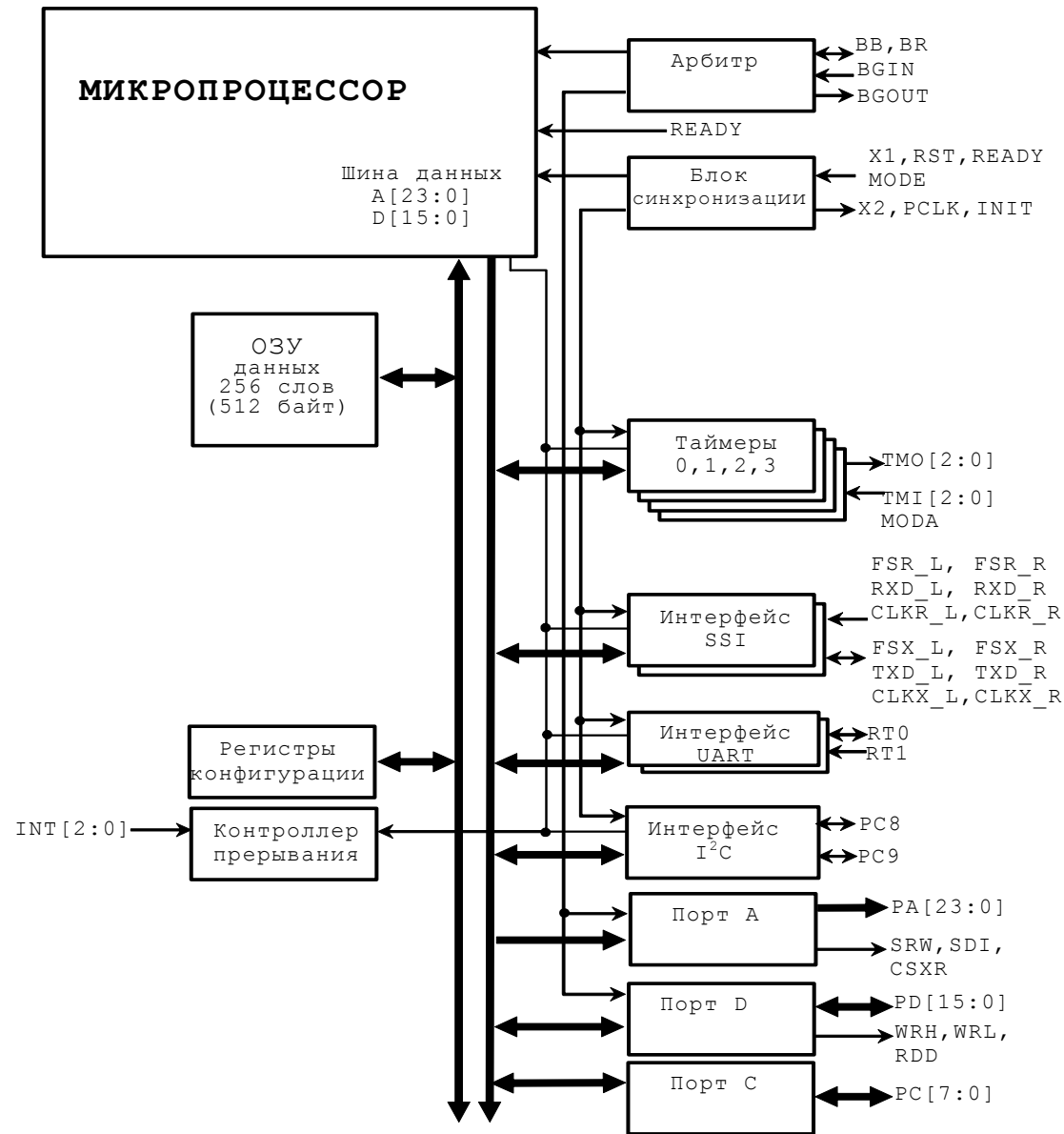


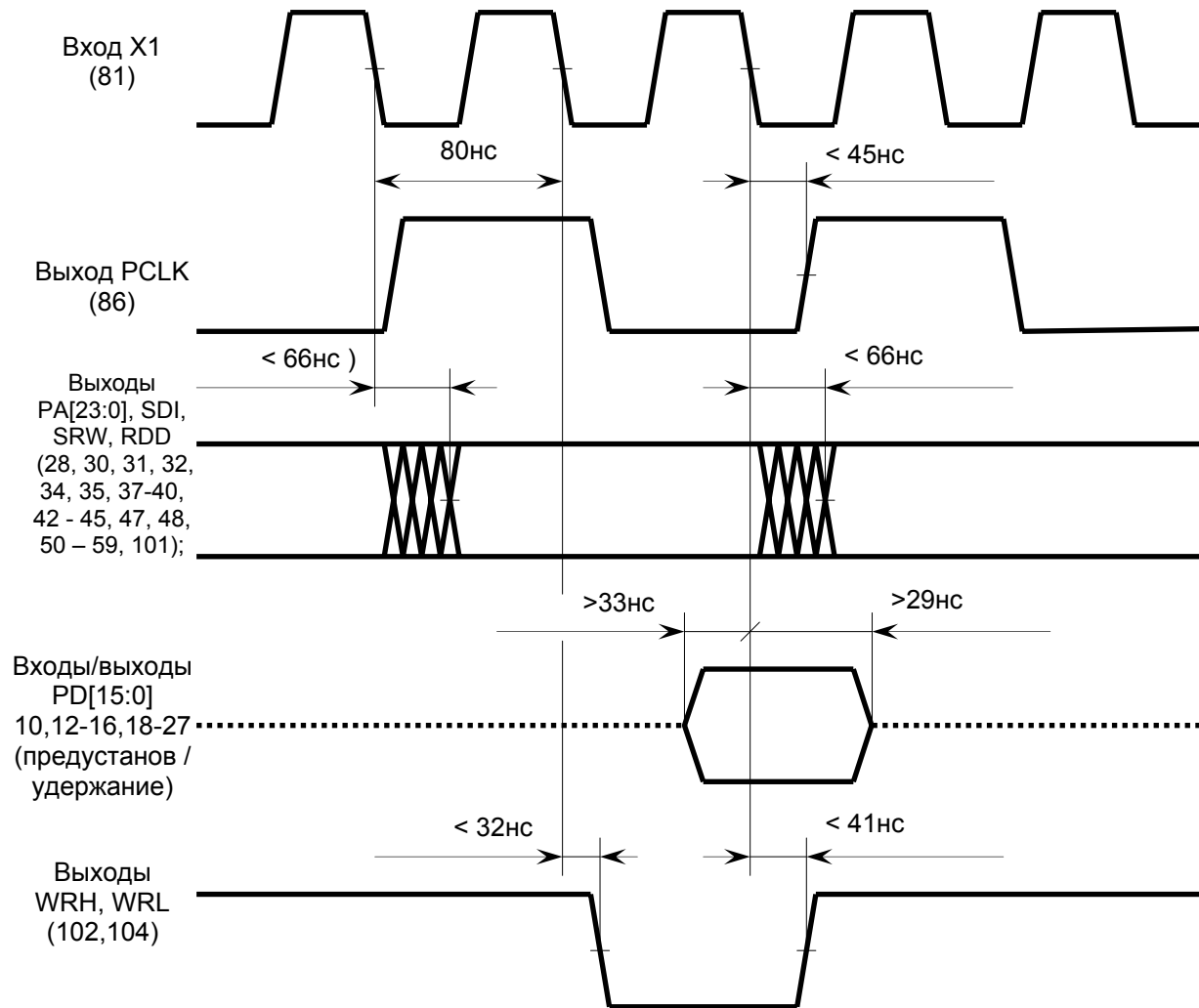
Рисунок 2 – Структурная схема микросхемы

Таблица 2 – Предельно-допустимые и предельные режимы эксплуатации микросхем

Наименование параметра режима, единица измерения	Буквенное обозначение	Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение питания, В	U_{CC}	4,5	5,5	-0,5	6,0
Входное напряжение низкого уровня, В	U_{IL}	0	0,8	-0,5	-
Входное напряжение высокого уровня, В	U_{IH}	$U_{CC}-0,8$	U_{CC}	-	6,0
Выходной ток низкого уровня, мА	I_{OL}	-	4	-	-
Выходной ток высокого уровня, мА	I_{OH}	-	$ -3 $	-	-
Частота следования импульсов тактовых сигналов, МГц	f_C	-	12,5	-	-
Емкость нагрузки, пФ	C_L	-	50	-	-

Таблица 3 – Электрические параметры

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Темпера- тура среды, ° С
		не менее	не более	
Выходное напряжение низкого уровня, В при $I_{OL}=4$ мА	U_{OL}	-	$\frac{0,40}{0,45}$	25 ± 10 -60 125
Выходное напряжение высокого уровня, В при $I_{OH}=-3$ мкА	U_{OH}	$U_{CC}-0,8$	-	
Входной ток низкого уровня, мкА при $U_{IL}=0$ В	I_{IL}	$\frac{-7}{-4}$	$\frac{-50}{-90}$	
Входной ток высокого уровня, мкА при $U_{IH}=U_{CC}$	I_{IH}	-	$\frac{10}{20}$	
Выходной ток низкого уровня в состоянии «Выключено», мкА при $U_{OL}=0$ В	I_{OZL}	$\frac{-7}{-4}$	$\frac{-50}{-90}$	
Выходной ток высокого уровня в состоянии «Выключено», мкА при $U_{OH}=U_{CC}$	I_{OZH}	-	$\frac{10}{20}$	
Ток потребления, мкА при $U_{IL}=0$ В, $U_{IH}=U_{CC}$	I_{CC}	-	$\frac{250}{500}$	
Динамический ток потребления, мА при $U_{IL}=0$ В, $U_{IH}=U_{CC}$, $f_C = 12,5$ МГц	I_{OCC}	-	$\frac{70}{100}$	



- Параметры уточняются в ходе ОКР

Рисунок 3 – Временная диаграмма работы микросхемы

Таблица 4 – Система команд

Назначение команды	Мнемоническое обозначение выполнения команды	Тактов вып. команды	Код команды					
			Слово 1					Слово 2
			[15-13]	12	[11-8]	[7-4]	[3-0]	
1 Загрузка адресного регистра операндом из команды	<регистр> = #;	2	000	0	Аaaa 23:20	Аaaa 19:16	rrrr	а...а 15:0
Загрузка регистра user	User = #;						0000	0 а...а
Сохранение rt в стеке , сохранение user в rt , загрузка регистра user операндом	User, <rt ; user = #;						0001	1 а...а
Загрузка регистра sp	sp = #;						0010	2 а...а
Загрузка регистра rp	rp = #;						0100	4 а...а
Загрузка регистра at	at = #;						1000	8 а...а
Сохранение at в стеке , загрузка регистра at	<at = #;						1001	9 а...а
Загрузка регистра rt	rt = #;						1100	C а...а
Сохранение rt в стеке , загрузка регистра rt	<rt = #;						1101	D а...а
Безусловный переход, глобальный	Jmp метка;						0110	6 а...а
Вызов подпрограммы, глобальный	<Имя п./п>();						0111	7 а...а
Вызов подпрограммы, в пределах текущей страницы в 4Кбайта	<Имя п./п>();	1		1	aaaa	aaaa	Aaaa	
2 Условный переход по адресу ,смещение, если (h – какой из стеков данных (ст/мл) используется при анализе) :	If(условие) <метка>;	1	001	h	cccc	aaaa	aaaa	
Знак операнда в регистре t равен нулю	If(!n) <метка>;				0000	0		
Знак операнда в регистре t равен единице	If(n) <метка>;				1000	8		
Регистр t не ноль	If(!z) <метка>;				0001	1		
Регистр t равен нулю	If(z) <метка>;				1001	9		
Значение v=(t[15]^psw[0]) равно нулю	If(!v) <метка>;				0010	2		
Значение v=(t[15]^psw[0]) равно единице	If(v) <метка>;				1010	A		
Бит переноса (c=psw[0]) равен нулю	If(!c) <метка>;		0		0011	3		
Бит переноса (c=psw[0]) равен единице	If(c) <метка>;		0		1011	B		
Для беззнаковых чисел A-B: A>B	If(!z)c или > <метка>;				0100	4		
Для беззнаковых чисел A-B: A<=B	If(c)z или <= <метка>;				1100	C		
Значение бита t[7] равно нулю	If(!nb) <метка>;				0110	B		
Значение бита t[7] равно единице	If(nb) <метка>;				1110	E		
Содержимое счётчика СТ не равно нулю	If(ct) <метка>;		0		0101	5		
Содержимое счётчика СТ ноль	If(!ct) <метка>;		0		1101	D		
Содержимое счётчика СТ не ноль, последующее вычитание из него 1	If(ct-) <метка>;		1		0101	5		
Содержимое счётчика СТ ноль, последующее вычитание из него 1	If(!ct-) <метка>;		1		1101	D		
Безусловный переход	If() <метка>;		0		0111	7		
Защикливание, до 7-ми команд	DO() <метка конца цикла>;		1		1111	f		

Техническая спецификация 1881 ВЕ2Т

Продолжение таблицы 4

Назначение команды	Мнемоническое обозначение выполнения команды	Такт ов вып. ком-ды	Код команды					
			Слово 1					Слово 2
			[15-13]	1 2	[11- 8]	[7 - 4]	[3 - 0]	
3 Работа с адресными регистрами, бит [0] d: «0» прм регистр– rt, «1» – at, стеки включены.	прм = reg + _;	1	010				Sss d	
Сложение с 8-ми разрядным смещением	<прм = источник + #			a	aaaa	0 aaa		
Сложение с 16-ти разрядным смещением из регистра t (знаковое)	<прм = источник + T;			h		1 000		
- // - , производится операция со стеком данных	<прм = источник + T>;					1 100		
Сложение с 16-ми разрядным смещением	<прм = источник + #;	2				1 001		d...d
Только для sp и gr: сложение текущего значения с #, запись р-га в регистр	Источник = источник + #	2				1 101		d...d
Источник – sp.	<прм = sp + _;						001 d	
Источник – gr.	<прм = gr + _;						010 d	
Источник – T.	<прм = T + _;						011 d	
Источник – at, не производится операций над стеком (замещение).	Прм = at + _;						100 d	
Источник – at.	<прм = at + _;						101 d	
Источник – rt, не производится операций над стеком (замещение).	Прм = rt + _;						110 d	
Источник – rt.	<прм = rt + _;						111 d	
4 Операции со стеками адресов и адресов возврата, номер регистра в стеке всегда вычисляется относительно текущего указателя стека со знаковым смещением ssss [11-8]. Сумма вычисляется с усечением результата до длины указателя стека (4 бита для стека RS, 3 бита для AS).		1	010		ssss			
В регистр стека возвратов заносится значение rt	R[rsp + #] = rt;					1-10	0--0	
В регистр стека адресов заносится значение at,	A[asp + #] = at;						0--1	
В rt заносится значение из стека адресов возврата	Rt = R[rsp + #];						1-00	
В at заносится значение из стека адресов	At = A[asp + #];						1-01	
5 Чтение из стека с 16-ти разрядным смещением	Рег. = Стек [указ.+#] + #V	2			ssss	1-11		v...v
В at записать значение из стека AS по адресу [указ.+#] увеличенное на #V	At = AS[#] + #V						1001	9 v...v
Тоже, но с запоминанием предыдущего значения в стеке	<at = AS[#] + #V						1011	B v...v
В at записать значение из стека RS по адресу [указ.+#] увеличенное на #V	At = RS[#] + #V						1101	D v...v
Тоже, но с запоминанием предыдущего значения в стеке	<at = RS[#] + #V						1111	F v...v
В rt записать значение из стека AS по адресу [указ.+#] увеличенное на #V	Rt = AS[#] + #V						1000	8 v...v
Тоже, но с запоминанием предыдущего значения в стеке	<rt = AS[#] + #V						1010	A v...v
В rt записать значение из стека RS по адресу [указ.+#] увеличенное на #V	Rt = RS[#] + #V						1100	C v...v
Тоже, но с запоминанием предыдущего значения в стеке	<rt = RS[#] + #V						1110	e v...v

Техническая спецификация 1881 ВЕ2Т

Продолжение таблицы 4

Назначение команды	Мнемоническое обозначение выполнения команды	Такто в вып. ком-ды	Код команды					
			Слово 1					Слово 2
			[15-13]	12	[11-8]	[7-4]	[3-0]	[15-0]
6 Специальные команды		1	011		-000	-00-		
Округление старшей половины регистра T[31:16] (подсуммирование T[15])	Round						000-	
Программное прерывание	Swi						001-	
Запрещение прерываний	Di						0100	4
Разрешение прерываний	Ei						0111	7
Выбор младшего байта	Byte l						1000	8
Выбор старшего байта	Byte h						1001	9
Останов	Wait						1010	a
Ожидание	Stop						1011	b
7 Операция АЛУ с регистровой пересылка между t и регистрами Бит [5] - g устанавливает формат t: '0' - 16 бит, '1' - 32 бита. 16 разрядное слово коммутируется битом [12] - h: '0' - t[15:0] (младшая), '1' - t[31:16] (старшая);	Операция АЛУ	1	011	h	fff	cccc	rrrr	
Пересылка t	Nop;				0000	0		
Сложение	t+n;				0001	1		
Обратное вычитание	n-t;				0010	2		
Вычитание	t-n;				0011	3		
Логическое поразрядное сложение «ИЛИ»	t n;				0100	4		
Логическое поразрядное умножение «И»	T&n;				0101	5		
Поразрядное «исключающее ИЛИ»	t^n;				0110	6		
Пересылка	n;				0111	7		
Поразрядная инверсия	~t;				1000	8		
Сложение с учётом переноса	t+n+c;				1001	9		
Обратное вычитание с учётом переноса	N+~t+c;				1010	A		
Вычитание с учётом переноса	t+~n+c;				1011	B		
Умножение кодов	U t*n;				1100	C		
Знаковое умножение	t*n;				1101	D		
Инверсное «исключающее ИЛИ»	~(t^n);				1110	E		
Пересылка с инверсией	t=~n;				1111	f		
Регистр user							0000	0
Регистр pc							0001	1
Регистр work							0010	2
Регистр sp							0011	3
Регистр swt (только для чтения)							0100	4
Регистр gp							0101	5
Регистр rt							0110	6
Регистр rt > + выгружается значение из стека в rt							0111	7
Регистр at							1000	8
Регистр at > + выгружается значение из стека в at							1001	9
Регистр xmas (только для записи)							1010	A
Регистр psw (r/w), xmas (только для чтения)							1011	B
Регистр buf							1100	C

Техническая спецификация 1881 ВЕ2Т

Продолжение таблицы 4

Назначение команды	Мнемоническое обозначение выполнения команды	Такт ов вып. ком-ды	Код команды							
			Слово 1						Слово 2	
			[15-13]	12	[11-8]	[7-4]	[3-0]	[15-0]		
Операция АЛУ – результат в n, операнд из регистра пересылается в t (16-ти разр., смотри h)	n = t-op-n; t = <регистр>;					0100	4			
Операция АЛУ – результат в n, операнд из регистра пересылается в T (32-х разр.)	n = t-op-n; T = <регистр>;					0110	6			
Операция АЛУ – результат в n сохраняется в стеке, операнд из регистра пересылается в t(16)	< n = t-op-n; t = <регистр>;					1100	c			
Операция АЛУ – результат в n сохраняется в стеке, операнд из регистра пересылается в T(32)	< n = t-op-n; T = <регистр>;					1110	e			
Операнд из t пересылается в регистр, операция АЛУ - результат в n	<регистр> = T; t = t-op-n;					01-1				
Операнд из t пересылается в регистр, операция АЛУ - результат в n сохраняется в стеке	<регистр> = T; t = t-op-n<;					11-1				
Операция короткого литерала и регистра t, результат в t	t = t -op- (m * 2 ^c)	1				00ee		mmmm		
Сохранение t в стеке, операция короткого литерала и регистра t, результат в t	< t = t -op- (m * 2 ^c)					10ee		mmmm		
Короткий литерал формата e='2' m='111-' не использовать, заменять длинным лит.										
Операция длинного литерала и регистра t, результат в t	t = # -op- t	2				0010	2	111-	d..d	
Сохранение t в стеке, операция длинного литерала и регистра t, результат в t	< t = # -op- t					1010	a	111-	d..d	
8 Одновременное выполнение двух операций: операции АЛУ - сдвига вторая – с регистрами или аппаратное умножение. Возможны любые комбинации операций АЛУ с операцией 2.		Формат операции 1	Операция 2	1	100	h	ffff shift(t)	Код формата 1	Код 2	
Операция АЛУ, результат в t	Увеличить sp на l	t = t-op-n;	sp++;				0000	0	0010	2
Операция АЛУ, результат в t, выгрузка из стека в n нового операнда	Уменьшить sp на l	t = t-op-n<;	sp--;				0100	4	0011	3
Операция АЛУ, результат в t, пересылка из t в n	Увеличить gr на l	t=t-op-n; n=t;	rp++;				1000	8	0100	4
Сохранение t в стеке, операция АЛУ, результат в t	Уменьшить gr на l	< t=t-op-n;	rp--;				1100	C	0101	5
Операция сдвига (см. ниже)	Уменьшить счётчик ct на l	t=shift(t);	ct--;				0001	1	0111	7
Операция сдвига, выгрузка из стека в n	Шаг умножения	t=shift(t); n<;	step *;				0101	5	1000	8
Операция сдвига, пересылка из t в n	Шаг деления	t=shift(t); n=t;	step _/;				1001	9	1001	9
Сохранение t в стеке, операция сдвига	Снятие стека в at	< t=shift(t);	at>;				1101	D	101-	
Операция АЛУ, результат в n, умножение старшей и младшей части регистра WORK, результат в t (при h=1 результат в T и wl=buf)		n=t-op-n;					0010	2		
Операция АЛУ, результат в n, выгрузка из стека в n, умножение старшей и младшей части регистра WORK, результат в t (при h=1 результат в T и wl=buf)		n=t-op-n>;					0110	6		
Сохранение n в стеке, операция АЛУ, результат в n, умножение старшей и младшей части регистра WORK, результат в t (при h=1 результат в T и wl=buf)		< n=t-op-n;					1110	E		
Операция АЛУ, результат в t	Возврат из подпрограммы	t=t-op-n;	exit;				0011	3	1100	C
Операция АЛУ, результат в t, выгрузка из стека в n	Возврат из прерывания	t=t-op-n<;	rti;				0111	7	1101	D
Операция АЛУ, результат в t, пересылка из t в n	Перенос rt в user и выгрузка стека в rt	t=t-op-n; n=t;	user=rt<;				1011	B	1110	E
Сохранение t в стеке, операция АЛУ, результат в t	Выгрузка информации из стека в rt	<t=t-op-n;	Rt<;				1111	f	1111	f

Продолжение таблицы 4

Назначение команды	Мнемоническое обозначение выполнения команды	Такто в вып. ком-ды	Код команды					
			Слово 1				Слово 2	
			[15-13]	12	[11- 8]	[7 - 4]	[3 - 0]	[15 - 0]
коды сдвигов в регистре T (32разряда) или младшей половине t (16 бит)	shift(t);				Код сдвига			
Логический сдвиг вправо (32 разряда)	0>>T;		0	0000	0			
Циклический сдвиг с учётом переноса вправо (32 разряда)	c>>T;		0	0001	1			
Циклический сдвиг вправо (32 разряда)	T0>>T;		0	0010	2			
Арифметический сдвиг вправо (32 разряда)	s>>T;		0	0011	3			
Логический сдвиг вправо (16 разрядов)	0>>t;		-	0100	4			
Циклический сдвиг с учётом переноса вправо (16 разрядов)	c>>t;		-	0101	5			
Циклический сдвиг вправо (16 разрядов)	t0>>t;		-	0110	6			
Арифметический сдвиг вправо (16 разрядов)	s>>t;		-	0111	7			
Логический сдвиг влево (32 разряда)	T<<0;		0	1000	8			
Циклический сдвиг с учётом переноса влево (32 разряда)	T<<c;		0	1001	9			
Циклический сдвиг влево (32 разряда)	T<<T0;		0	1010	A			
Арифметический сдвиг влево (32 разряда)	T<<s;		0	1011	B			
Логический сдвиг влево (16 разрядов)	t<<0;		-	1100	C			
Циклический сдвиг с учётом переноса влево (16 разрядов)	t<<c;		-	1101	D			
Циклический сдвиг влево (16 разрядов)	t<<t0;		-	1110	E			
Арифметический сдвиг влево (16 разрядов)	t<<s;		-	1111	F			
9 Операции с чтением данных из памяти с косвенной адресацией через регистр «reg»		1	101	h	ffff shift	cccc	rrrr	
	Режимы адресации							
Адрес в регистре sp	sp						0000	0
Адрес в регистре gr	gr						0001	1
Адрес в регистре sp, значение sp увеличивается на 1	sp++						0010	2
Адрес в регистре sp, значение sp уменьшается на 1	sp—						0011	3
Адрес в регистре gr, значение gr увеличивается на 1	gr++						0100	4
Адрес в регистре gr, значение gr уменьшается на 1	gr—						0101	5
Адрес в регистре T	T						0110	6
Адрес в счётчике команд, значение увеличивается на 1	pc++						0111	7
Адрес в регистре at, значение gr увеличивается на 1	at++						1000	8
Адрес в регистре at, значение gr уменьшается на 1	at—						1001	9
Адрес в регистре at	at						1010	A
Адрес в регистре at, в at выгружается значение из стека	At<						1011	B
Адрес в регистре rt, значение rt увеличивается на 1	rt++						1100	C
Адрес в регистре irt, значение irt увеличивается на 1 – обращение к памяти программ	irt++	2					1101	D
Адрес в регистре rt	Rt	1					1110	E
Адрес в регистре rt, в rt выгружается значение из стека	Rt<						1111	F

Техническая спецификация 1881 ВЕ2Т

Продолжение таблицы 4

Назначение команды	Мнемоническое обозначение выполнения команды	Такто в вып. ком-ды	Код команды					
			Слово 1					Слово 2
			[15-13]	1 2	[11- 8]	[7 - 4]	[3 - 0]	[15 - 0]
	Формат операции							
АЛУ на младшей половине стека данных: результат в n , чтение в t	n = t-op-n; t = reg;				0000	0		
Сохранение в стеке данных n , АЛУ на младшей половине стека данных: результат в n , чтение в t	< n = t-op-n; t = reg;				1000	8		
АЛУ на младшей половине стека данных результат в n , перемножение старшей и младшей частей work, результат в t , чтение в buf	n = t-op-n; t = wl * wh; buf = reg;		0		0010	2		
Сохранение в стеке данных n , АЛУ на младшей половине стека данных результат в n , перемножение старшей и младшей частей work, результат в t , чтение в buf	< n = t-op-n; t = wl * wh; buf = reg;		0		1010	A		
АЛУ на старшей половине стека данных результат в n , перемножение старшей и младшей частей work, результат в T (32 разр.) , чтение в старшую половину регистра work , загрузка значения регистра buf в младшую половину регистра work	%; n = t-op-n; T = wl * wh; wh = reg; wl = buf;		1		0010	2		
Сохранение в стеке данных n , АЛУ на старшей половине стека данных результат в n , перемножение старшей и младшей частей work, результат в T (32 разр.) , чтение в старшую половину регистра work , загрузка значения регистра buf в младшую половину регистра work	%; < n = t-op-n; T = wl * wh; wh = reg; wl = buf;		1		1010	A		
Сдвига, результат в n (N) , чтение в t	n = shift(t); t = reg;				0001	1		
Сохранение в стеке данных n , сдвига, результат в n (N) , чтение в t	< n = shift(t); t = reg;				1001	9		
АЛУ на младшей половине стека данных результат в t , чтение в buf	t = t-op-n; buf = reg;		0		0011	3		
АЛУ на младшей половине стека данных результат в t , выгрузка из стека данных в n , чтение в buf	t = t-op-n<; buf = reg;		0		1011	B		
АЛУ на старшей половине стека данных результат в t , чтение в старшую половину регистра work , загрузка значения регистра buf в младшую половину регистра work	%; t = t-op-n; wh = reg; wl = buf;		1		0011	3		
АЛУ на старшей половине стека данных результат в t , выгрузка из стека данных в n , чтение в старшую половину регистра work , загрузка значения регистра buf в младшую половину регистра work	%; t = t-op-n<; wh = reg; wl = buf;		1		1011	B		
10 Операции и чтение данных с косвенной и постоянным 5 битовым смещением в литерале. Адрес: user[23:5].r[4:0] (r[4] - бит[13] команды, r[3:0] - биты [3:0])	Регистр = rg[« адрес »];		11a	h	ffff shift	Cccc	aaaa	

Продолжение таблицы 4

Назначение команды	Мнемоническое обозначение выполнения команды	Такто в вып. ком-ды	Код команды					
			Слово 1					Слово 2
			[15-13]	1 2	[11- 8]	[7 - 4]	[3 - 0]	[15 - 0]
11 Сохранение регистра <i>t</i> в памяти данных (кода способа адресации определяется полем <i>grrr</i>) и операция определенная битами <i>ffff dddd</i>	<i>rg</i> [«адрес »] = <i>t</i> ; операция		101	h	ffff shift	Dddd	гггг	
АЛУ на младшей половине стека данных результат в <i>t</i>	<i>reg</i> = <i>t</i> ; <i>t</i> = <i>t-op-n</i> ;					0100		
АЛУ на младшей половине стека данных результат в <i>t</i> , выгрузка из стека данных в <i>n</i>	<i>reg</i> = <i>t</i> ; <i>t</i> = <i>t-op-n</i> <;					1100		
АЛУ на младшей половине стека данных результат в <i>n</i> , перемножение старшей и младшей частей <i>work</i> , результат в <i>t</i>	<i>reg</i> = <i>t</i> ; <i>n</i> = <i>t-op-n</i> ; <i>t</i> = <i>wl</i> * <i>wh</i> ;		101	0	ffff shift	0110	гггг	
Сохранение в стеке данных <i>n</i> , АЛУ на младшей половине стека данных результат в <i>n</i> , перемножение старшей и младшей частей <i>work</i> , результат в <i>t</i>	<i>reg</i> = <i>t</i> ; < <i>n</i> = <i>t-op-n</i> ; <i>t</i> = <i>wh</i> * <i>wl</i> ;			0		1110		
АЛУ на старшей половине стека данных результат в <i>n</i> , перемножение старшей и младшей частей <i>work</i> , результат в <i>T</i> (32 разр.)	%; <i>reg</i> = <i>t</i> ; <i>n</i> = <i>t-op-n</i> ; <i>T</i> = <i>wh</i> * <i>wl</i> ; <i>wl</i> = <i>buf</i> ;			1		0110		
Сохранение в стеке данных <i>n</i> , АЛУ на старшей половине стека данных результат в <i>n</i> , перемножение старшей и младшей частей <i>work</i> , результат в <i>T</i> (32 разр.) , загружается значение из регистра <i>buf</i> в старшую половину регистра <i>work</i>	%; <i>reg</i> = <i>t</i> ; < <i>n</i> = <i>t-op-n</i> ; <i>T</i> = <i>wh</i> * <i>wl</i> ; <i>wl</i> = <i>buf</i> ;			1		1110		
Сдвига, результат в <i>t</i> (<i>T</i>)	<i>reg</i> = <i>t</i> ; <i>t</i> = <i>shift</i> (<i>t</i>);			h		0101		
Сдвига, результат в <i>t</i> (<i>T</i>), выгрузка из стека данных в <i>n</i>	<i>reg</i> = <i>t</i> ; <i>t</i> = <i>shift</i> (<i>t</i>); <i>n</i> <;					1101		
АЛУ результат в <i>t</i>	<i>reg</i> = <i>t</i> ; <i>t</i> = <i>t-op-n</i> ;					0111		
АЛУ результат в <i>t</i> , выгрузка из стека данных в <i>n</i>	<i>reg</i> = <i>t</i> ; <i>t</i> = <i>t-op-n</i> >;					1111		
12 Сохранение <i>t</i> по адресу <i>user</i> [23:5]. <i>r</i> [4:0] - (<i>r</i> [4] - бит[13] команды, <i>r</i> [3:0] - [3:0]) и операции, определяемые полями <i>ffff dddd</i> , из перечисленных выше	<i>rg</i> [«адрес »] = <i>t</i> ; операция		11a	h	ffff shift	dddd	aaa a	