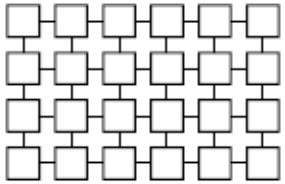
"Матричные микромощные мультикомпьютеры".

- обзор архитектуры процессоров GreenArrayChips - GA144;
- структура и система команд вычислительных ядер F18A;
- топология мультипроцессоров;
- периферийные устройства;
- встроенное программное обеспечение;
- среда ArrayForth;
- перспективы разработок на базе GA144.

История стековых MISC мультипроцессоров

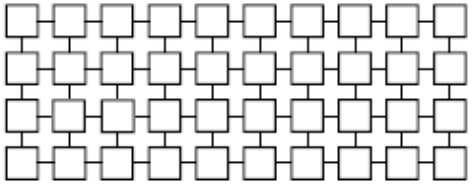
IntellaSys

SEAforth24



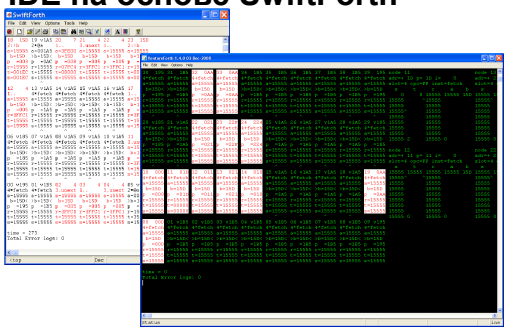
2007

SEAforth40



2008

IDE на основе SwiftForth

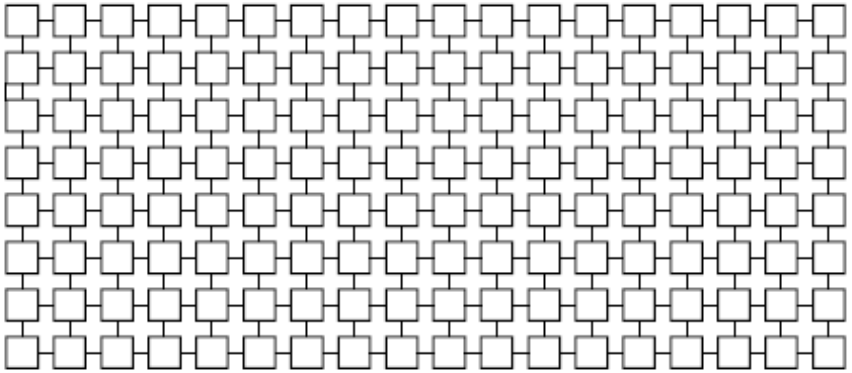


GreenArrayChips

GA4

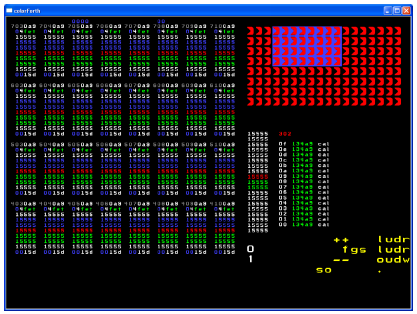
2009-2010

GA144



2010-2011

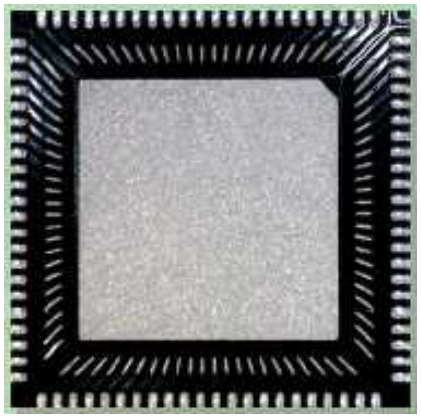
IDE на основе colorForth



144-ядерные процессоры GreenArray - GA-144

Общая информация

- размеры - 1x1 см (корпус QFN88);



- технология - 180 нм;

- производительность отдельного ядра 650-700 MIPS;

- суммарная пиковая производительность порядка 96 миллиардов операций в секунду;

- потребляемая мощность одним ядром ~ 6-10мВт (в режиме покоя около 100 нВт);

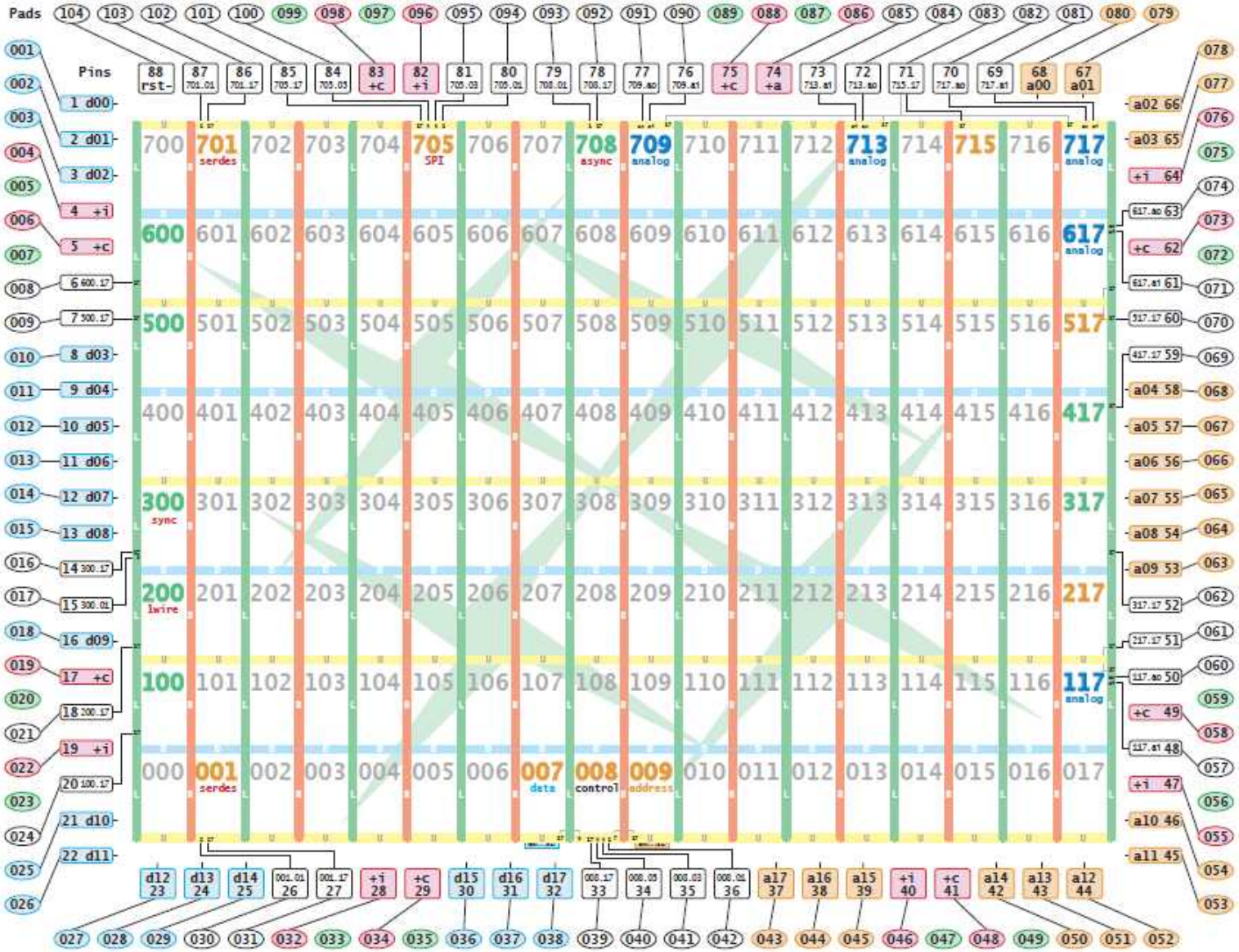
- суммарная потребляемая мощность 0,014 – 0,65 Вт;

-- напряжение питания 1,8 В.

http://www.greenarraychips.com/home/documents/greg/PB001-100503-GA144-1-10.pdf

Топология

- 144 ядра образуют решетку 18x8 с непосредственными связями между соседями;



http://www.greenarraychips.com/home/documents/bee/g144poster.A1.pdf

Ключевые особенности архитектуры

- однородная вычислительная среда на кристалле;

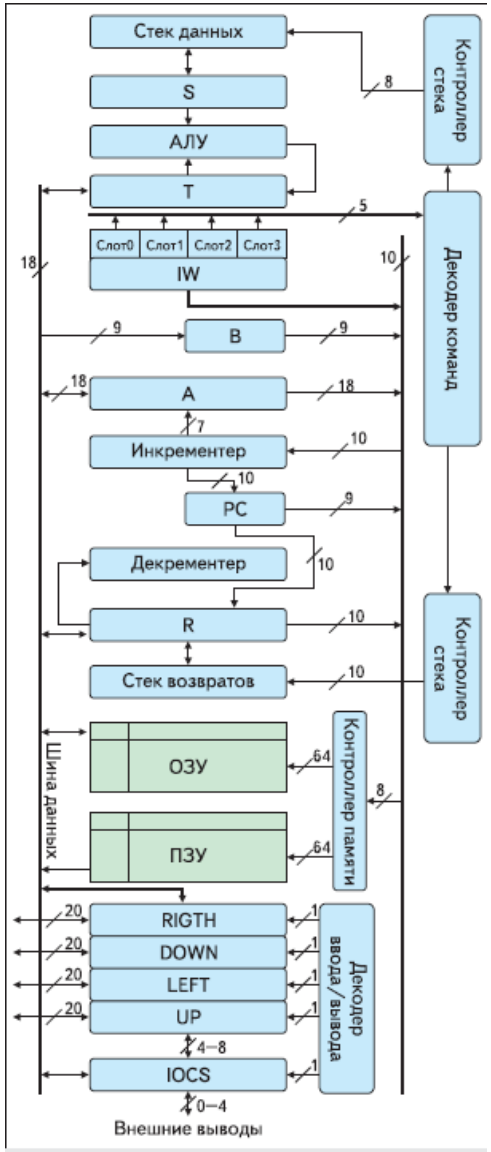
-основной подход к программированию - множество параллельных взаимодействующих процессов, синхронизирующихся при взаимодействии (теорией взаимодействующих последовательных процессов Ч. Хоара - ОККАМ) - наиболее близкий аналог из распространенных языков - параллельные процессы в АДА с механизмом рандеву;

-все ядра работают полностью асинхронно - по косвенным признакам можно отнести к самосинхронным – тактовые линии и генераторы отдельных ядер отсутствуют;

- синхронизация процессов на различных ядрах идет через коммуникационные порты - реализовано аппаратно – механизм блокировок ядер.

http://peter.sovietov.com/txt/seaforth/seaforth.pdf

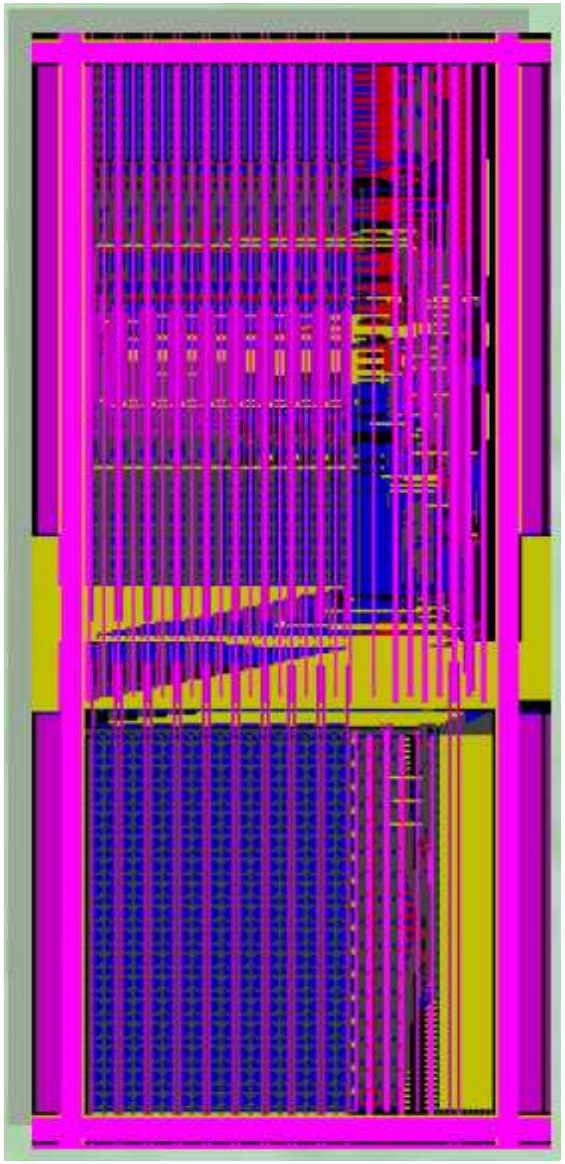
Ядра F18A



Структурная схема ядра процессора

-стековые 18-битные фон-Неймановские ядра;

- <20000 транзистров/ядро



структура кристалла ядра

Ядра F18A ***http://www.greenarraychips.com/home/doc ... 2-F18A.pdf***

Система команд

- 32 5-битных инструкции ;

Opcode	Hex	Notes -- ADDRESS opcodes		Opcode	Hex	Notes -- ALU opcodes
;	**00**	**return**		**+***	**10**	**. +***
ex	01	execute via r (swap p and r)		2*	11	left shift
name ;	02	jump to a red word, name		2/	12	right shift (signed)
name	03	call to a red word, name		-	13	invert (3ffff xor)
unext	**04**	**jump r≠0 decrement r**		**+**	**14**	**. +**
next	05	jump r≠0 decrement r		and	15	
if	06	jump t=0		or	16	exclusive or (xor)
-if	07	jump t17=0		drop	17	
@p	**08**	**literal**	**7-bit auto-increment**	**dup**	**18**	
@+	09	fetch via a	7-bit auto-increment	pop	19	
@b	0a	fetch via b		over	1a	
@	0b	fetch via a		a	1b	fetch from register a
!p	**0c**		**7-bit auto-increment**	**.**	**1c**	**nop**
!+	0d	store via a	7-bit auto-increment	push	1d	
!b	0e	store via b		b!	1e	store into register b
!	0f	store via a		a!	1f	store into register a

- упаковываются в слоты - 3 полных 5-ти битных, плюс 3-битный остаток;

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode Slot	Slot 0					Slot 1					Slot 2					Slot 3		

Временные затраты на выполнение инструкций

Тип инструкции	Описание типа	Время выполнения*
Арифметико-логические инструкции	Все арифметико-логические инструкции за исключением сложения	1 unit
	Сложение	1 или 2
Работа с памятью	Операции с регистром статуса IO	2
	Доступ к портам (если данные в коммуникационном порту готовы)	3
	Доступ к ОЗУ или ПЗУ	3,6
	Доступ к шинам данных или адреса	3,5
	Чтение счетчика АЦП	3,6

*** (1 unit ~ 1,5 нс)**
предварительные данные

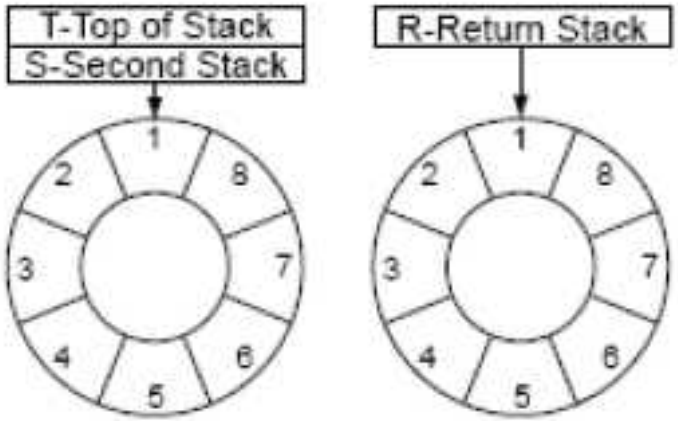
- Среднее время выполнения инструкции порядка 1,5 нс при затратах энергии 7пДж;

Регистровый состав

- стек данных глубиной 10 слов - последние 8 слов замкнуты в кольцо;
- стек возвратов глубиной 9 слов - последние 8 слов замкнуты в кольцо;

- 2 индексных регистра:
  - 18-битный регистр A;
  - 9-ти битный регистр B;

- регистр статуса IOR - отображает статус коммуникационных портов, используется для управления внешними выводами;



IO	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Read	PIN	Rr_	Rw	Dr_	Dw	Lr_	Lw	Ur_	Uw				PIN		PIN		PIN	
Write	PIN	state				DB	WD		pin		pin		PIN	state	PIN	state	PIN	state
	SR				VCO						9-bit DAC output level				(155 xor)			

PIN state	01=WEAK PULLDOWN	00=tristate	10=Vss(0)	11=Vdd(1)	-Pin State
WD	0=NORMAL	1=inverted			-Wake Direction
DB	1=OUTPUT	0=tristate			-Data Bus direction
SR	0=RECEIVE	1=send			-Serializer/Deserializer
VCO	10=OFF	00=input	01=Vdd calibrate	11=Vss calibrate	-Voltage Controlled Oscillator

- 10-битный регистр P (счетчик команд) 9-й бит — режим арифметики (обычный - перенос при арифметических операциях не учитывается и расширенный - перенос учитывается), 8-й бит — пространство ввода-вывода, 0-7 - адрес;

Карта памяти

- ОЗУ, ПЗУ, коммуникационные порты находятся в одном адресном пространстве;

Port	Address	Description
-d-u	105	Down, Up
-d--	115	Down
-dlu	125	Down, Left, Up
-dl-	135	Down, Left
data	141	Up, no handshake
---u	145	Up
io	15d	18-bit I/O Control/Status
--lu	165	Left, Up
--l-	175	Left

Port	Address	Description
rd-u	185	Right, Down, Up
rd--	195	Right, Down
rdlu	1a5	Right, Down, Left, Up
rdl-	1b5	Right, Down, Left
r--u	1c5	Right, Up
r---	1d5	Right
r-lu	1e5	Right, Left, Up
r-l-	1f5	Right, Left

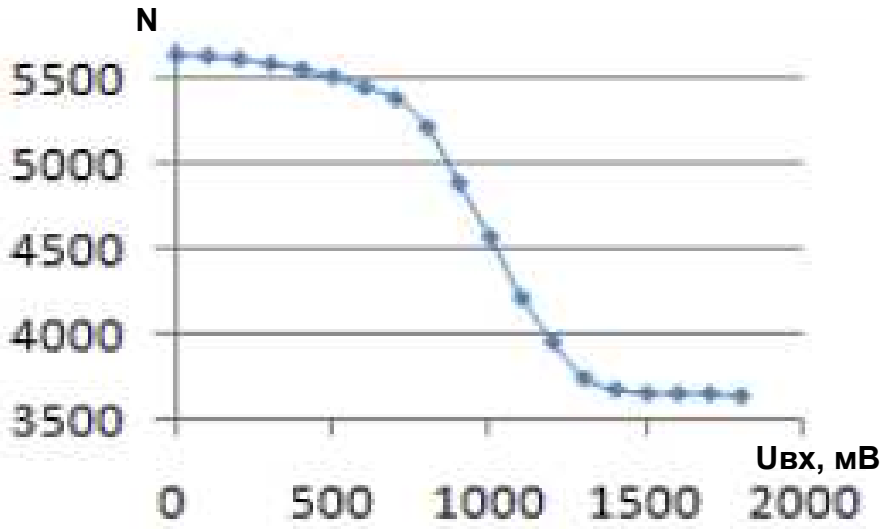
Note: ALWAYS refer to port names, addresses may change

- возможно исполнять инструкции из коммуникационного порта - переход на адрес одного из портов.

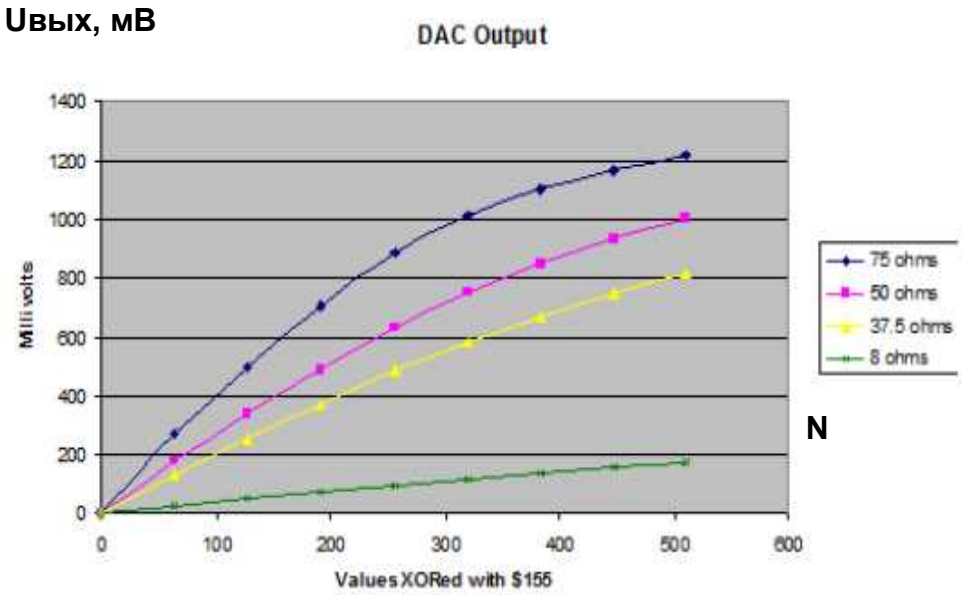
Периферийные устройства

- 22 крайних ядра решетки имеют одну и более линию ввода-вывода;
- от 1 до 4 GPIO;

-- 5 АЦП - построены по принципу ГУН+счетчик - в приложении есть возможность выбора - точность/скорость;



-- 5 ЦАП - 9 бит с токовым выходом - диапазон нагрузок от 4 до 100 Ом (75, 50, 37.5, 8 Ом);



-- SPI (интерфейс, прошитый в ПЗУ для 25й серии микросхем флеш-памяти);
-- 2 параллельных 18-битных порта (вход/выход/Z-состояние) - скорость до 20-30 Мслов/с;
-- 2 высокоскоростных последовательных интерфейса SERDES - данные+тактовый (скорость порядка 400 Мбит/с).

http://www.greenarraychips.com/home/doc ... 18A-IO.pdf

Встроенное программное обеспечение

Наборы слов ПЗУ							
Math rom (1)	SerDes (2)	Analog (3)	1-wire (4)	Sync serial (5)	SPI boot (6)	Async serial (7)	Null (8)

					--+		
			rcv		+--		
			bit		+-+		
					-++		
relay	relay	relay		relay	relay	relay	
warm	warm	warm	warm	warm	warm	warm	warm
	cold		cold	cold	cold	cold	
					8obits	wait	
multiply	multiply	multiply	multiply	multiply	ibit	sync	
fractional multiply	fractional multiply	fractional multiply	fractional multiply		half	start	
taps	*taps*			*taps*	select	delay	
interpolate	*interpolate*	*interpolate*	*interpolate*		obit	lsh	
triangle	triangle	triangle	triangle	triangle	rbit	rsh	
-u/mod	-u/mod	-u/mod	-u/mod		18ibits	18ibits	
		-dac		sget	u2/	4bits	
				6in	spispeed	2bits	
polynomial approximation		*polynomial approximation*		2in	spi-boot		
				ser-exec	spi-exec	ser-exec	
				ser-copy	spi-copy	ser-copy	

Арифметические функции

Математические подпрограммы

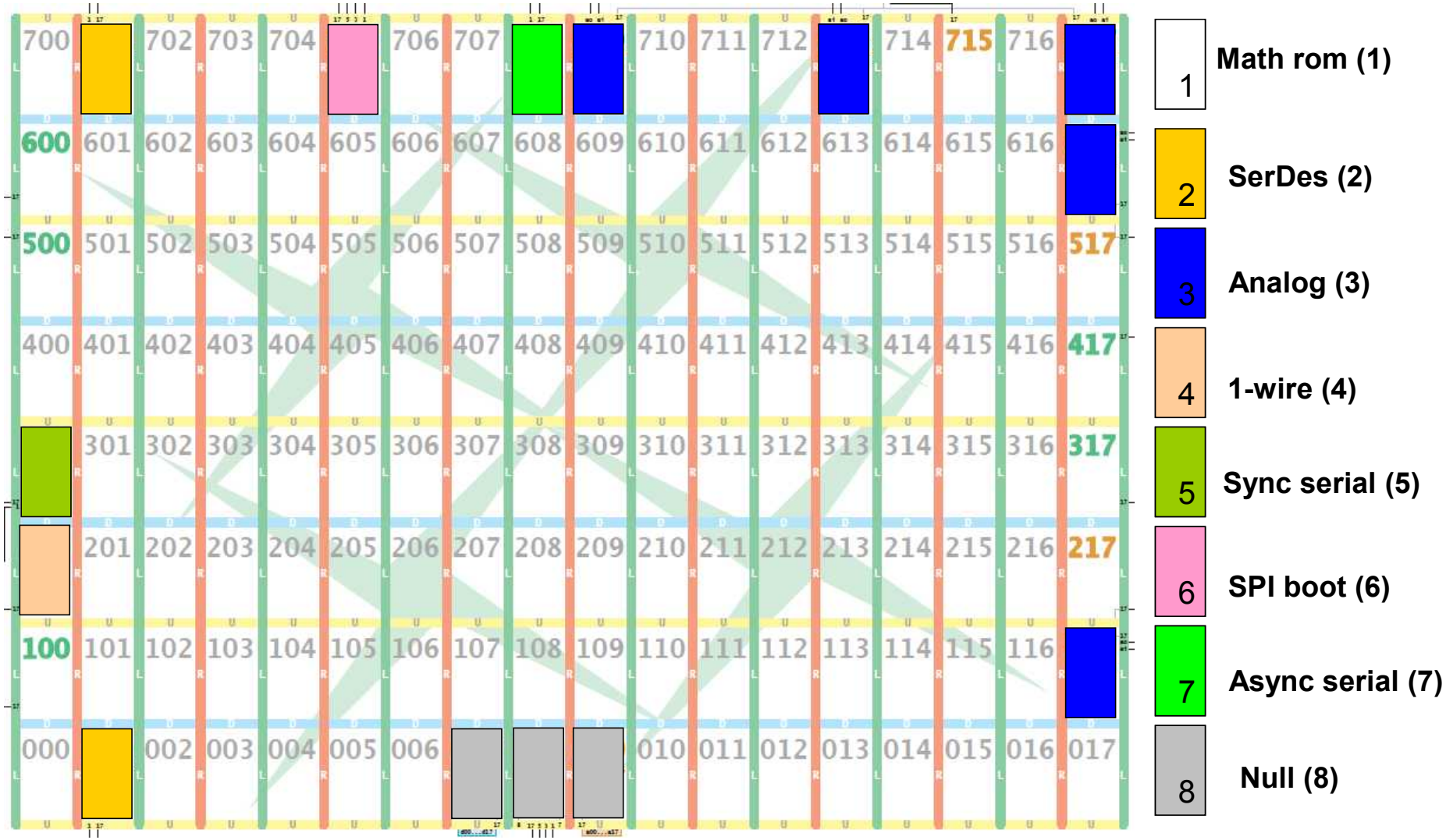
Инициализация кода ядра, перезагрузка

Работа с линиями последовательного ввода-вывода

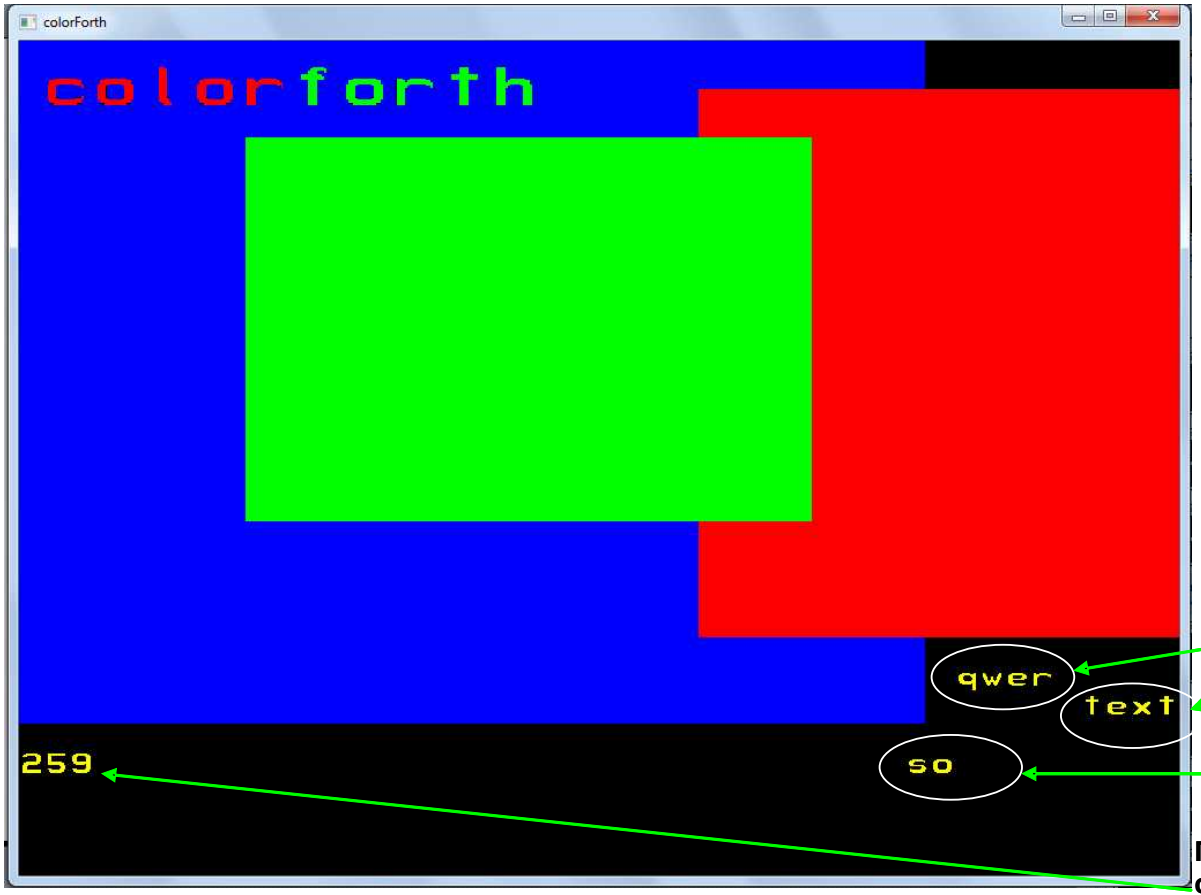
Слова среднего уровня для приема-передачи данных

Аналоговый ввод-вывод

Карта ПЗУ


700 702 703 704 706 707 710 711 712 714 715 716
600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616
500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517
400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217
100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116
000 002 003 004 005 006 010 011 012 013 014 015 016 017
1 Math rom (1)
2 SerDes (2)
3 Analog (3)
4 1-wire (4)
5 Sync serial (5)
6 SPI boot (6)
7 Async serial (7)
8 Null (8)

Среда разработки - ArrayForth



Режим работы клавиатуры (раскладка)
QWERTY или **dvorak**

Тип введенных данных число или текст

Вводимая строка

Представление вводимой строки в виде числа в текущей системе счисления

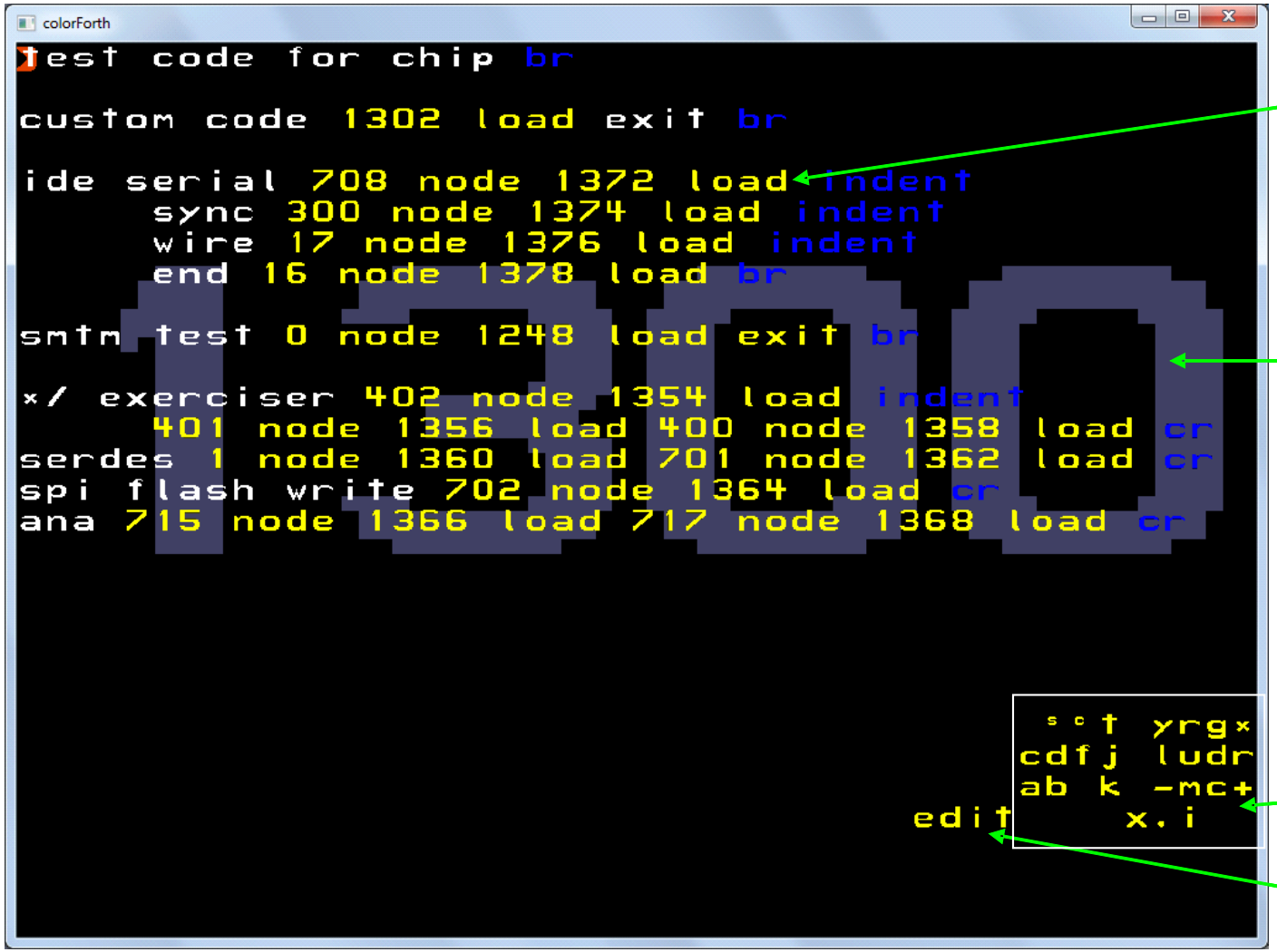
Основные файлы среды разработки (в версии для Windows):

Okad.bat - bat-файл с настройками среды;

Okad2-41-pd.exe - исполняемый файл среды для её запуска в системе Windows;

OkadWork.cf - рабочая память системы (загружаемый/сохраняемый образ).

Среда разработки – ArrayForth – блочный редактор



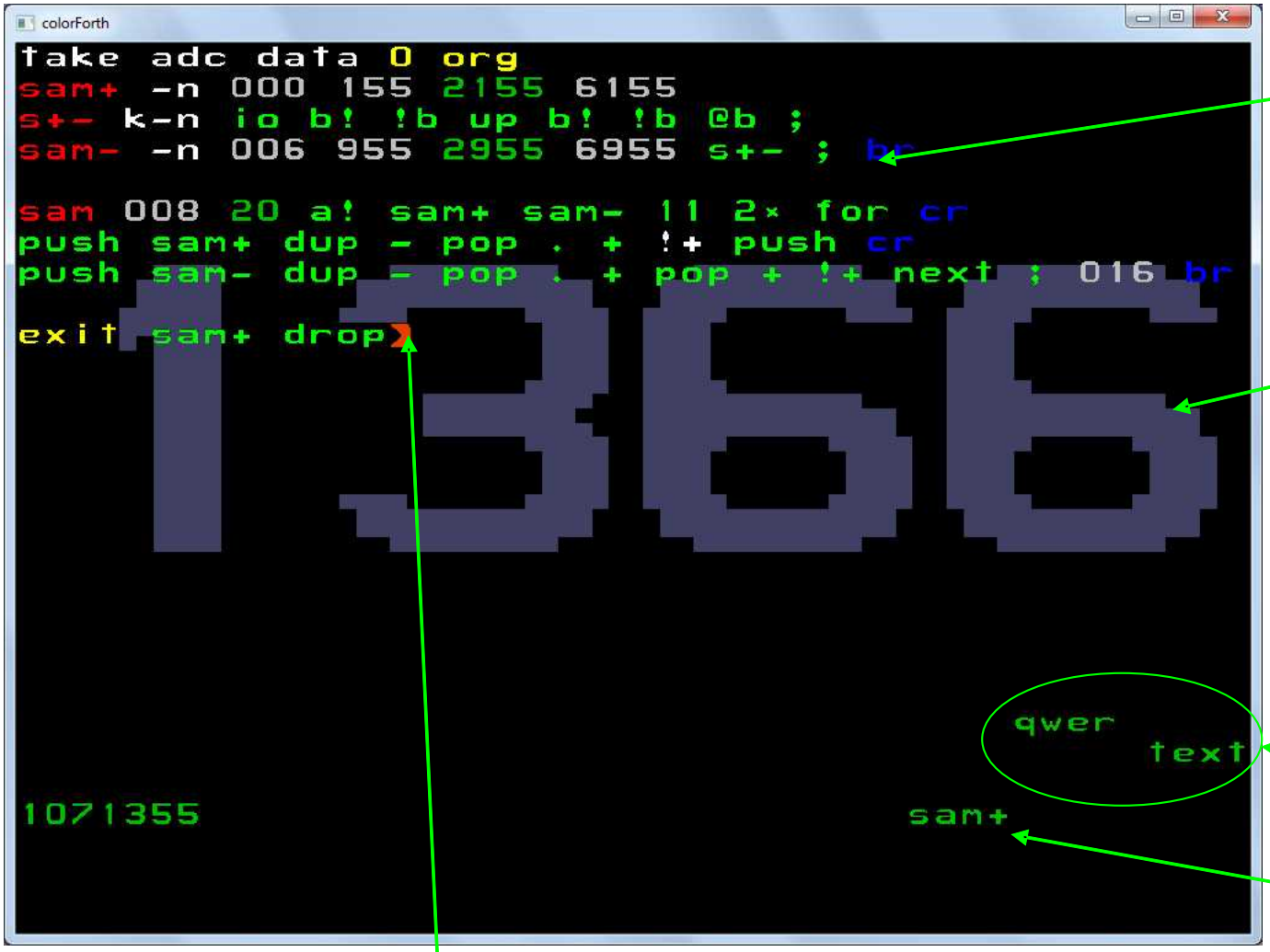
Текст блока

Номер блока

«Подсказки» команд редактора

Вводимое слово

Среда разработки – ArrayForth блочный редактор



Текст блока

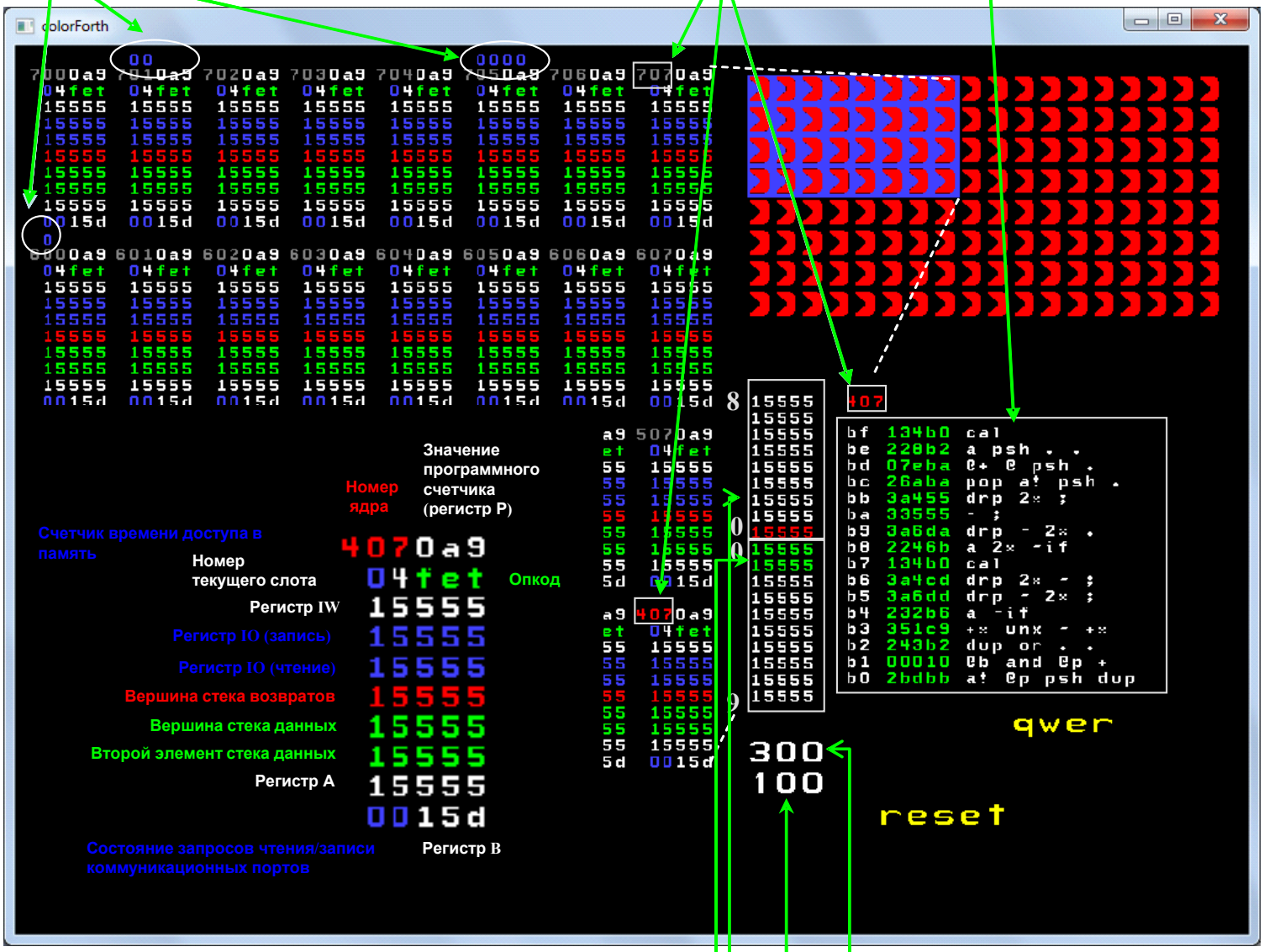
Номер блока

Текущий режим

Вводимое слово

Текущее положение курсора

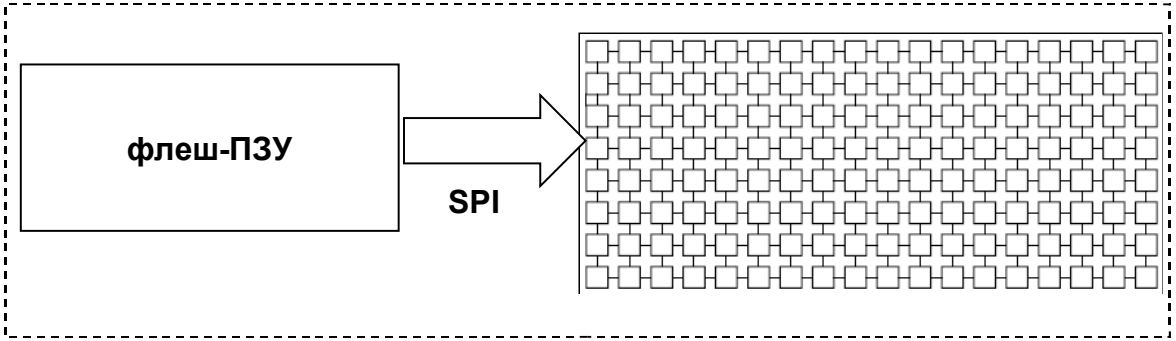
ArrayForth симулятор



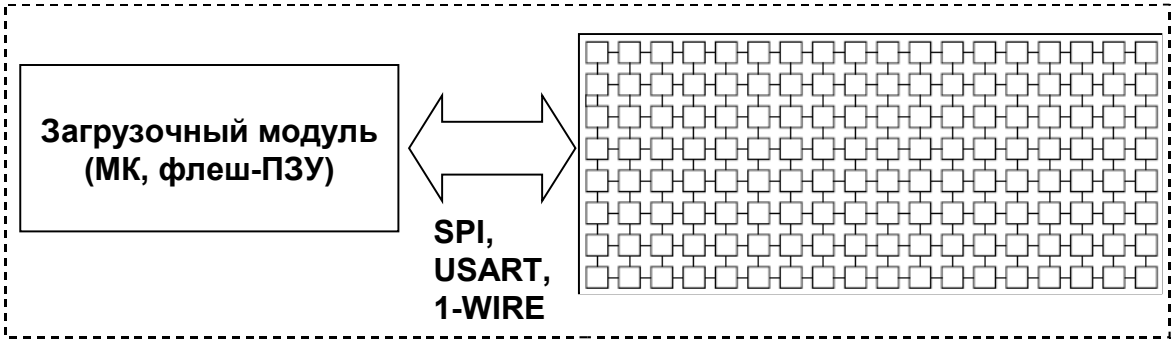
ArrayForth состояние ядер

Способы решения задач с использованием GA144

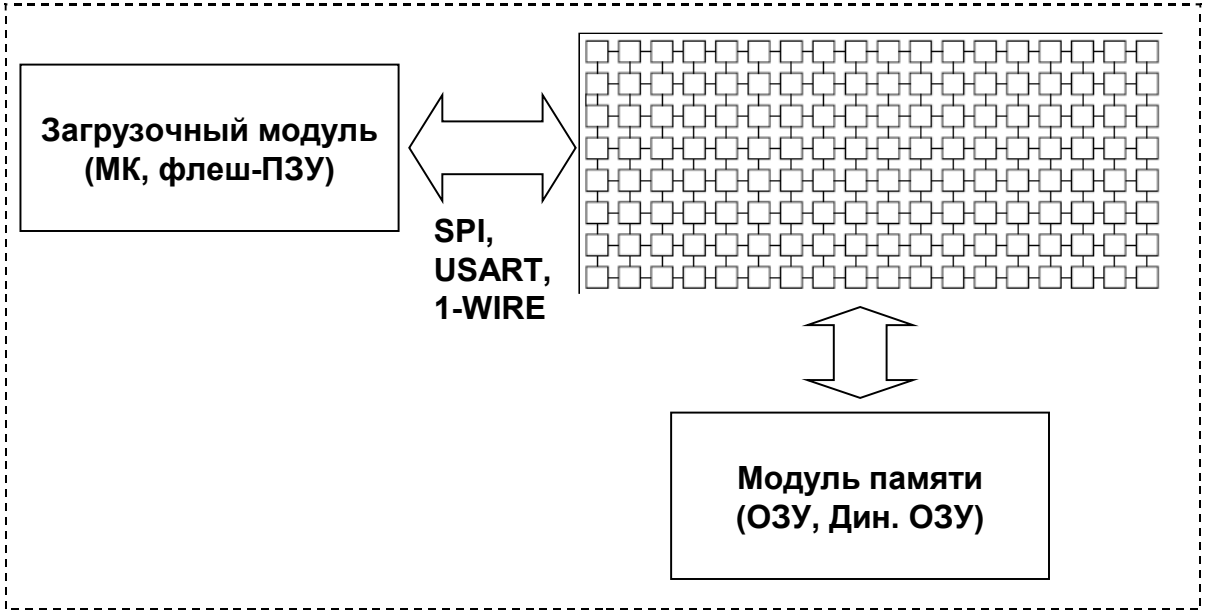
Выполнение одной программы, загружаемой в процессорные ядра при начальной инициализации



Смена исполнимого кода всех ядер или отдельных групп, инициированная внешним управляющим контроллером или программой самого процессора

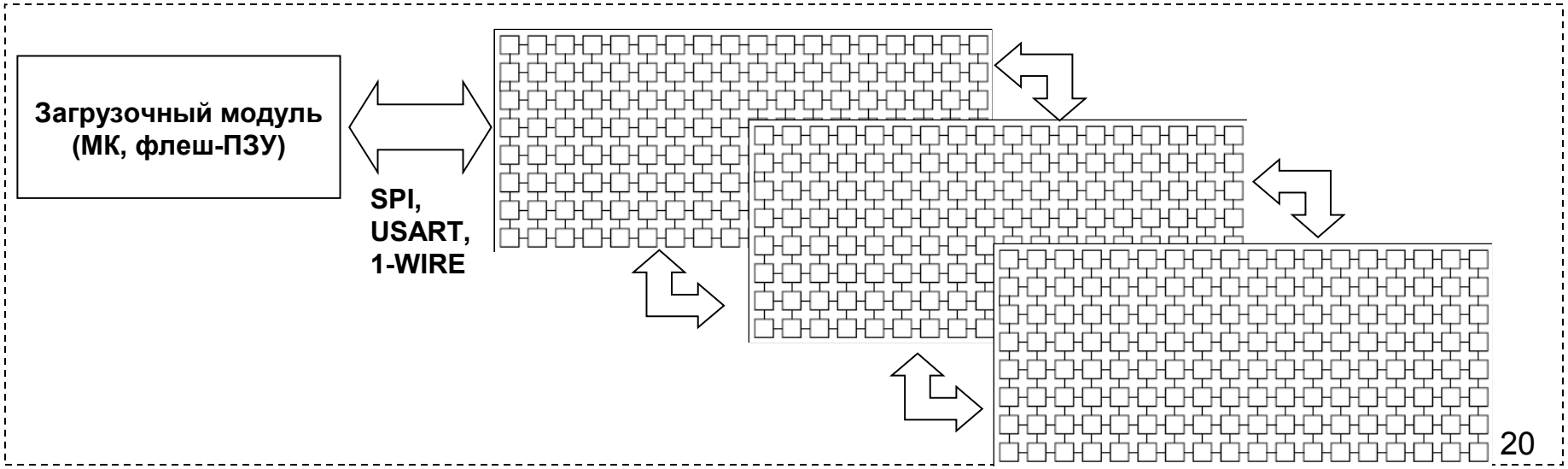


Способы решения задач с использованием GA144



Расширение оперативной памяти, создание ВМ, работающих на группах из нескольких ядер,

Наращивание количества процессоров, использование механизмов исполнения кода с коммуникационных портов



<u>Возможные целевые области применения GA144</u>

- робототехника -манипуляторы/протезы/автономные подвижные роботы;
- нейронные сети - классификация/распознавание сигналов/образов;
- «бортовые системы» - диагностика состояния в реальном времени/ контроль движения;
- «академические» системы - аппаратное обеспечение курсов цифровой обработки сигналов, параллельного программирования, архитектуры вычислительных систем;
- «персональные» вычислительные системы - расширения ПК/планшетники/ «гаджеты»;
- Java/LISP/Prolog – машины;
- распознавание/синтез речи;
- управление антенными системами (ЦАР, ФАР);
- модуляторы/демодуляторы сигналов.

Программные «трюки»

Swap (x y -- y x)

```
over ... drop
push a! pop a
over push push drop pop pop
over push over or or pop
```

Nip (x y -- y)

```
nip push drop pop ;
nip over or or ;
```

Zero (-- 0)

```
dup dup or
dup or
```

Subtract (x y -- x-y)

```
- 1 . + . +
```

```
3* dup 2* . + ;
5* dup 2* 2* . + ;
10* 2* 5* ;
```


http://www.greenarraychips.com/home/documents/greg/cf-code-library.htm

Программные «трюки»

«CASE»

```
jump n pop + push ;
example n jump zero ; one ; two ;
```

Сравнение чисел

```
min nn-n - over . + - -if + ; then drop ;
max nn-n - over . + - -if drop ; then + ;

min nn-n - over . + - -if begin + ;
max nn-n - over . + - -until then drop ;
```

VALUE-переменные

```
x! n @p drop !p ;
x -n 100 ;

!delay n @p drop !p ;
delay 100 for . . unext ;
```

Программные «трюки»

Цифровой фильтр

```
taps    yxc-y'x'    cr
for     example...  br

fir yx-y'x' 15 taps -53 , 0 , 2276 , 0 , 382 ,
0 , -1706 , 0 , -1158 , 0 , 2014 , 0 , 2406 ,
0 , -1977 , 0 , -4206 , 0 , 1289 , 0 , 6801 ,
0 , 678 , 0 , -11109 , 0 , -6250 , 0 , 23531 ,
0 , 54145 , 0 , br
```

Интерполяция значений

```
0 org 0 , 450 , 900 , 1350 , 1800 , cr
mv    i-n    3F    5    interp    ;    br

0      mv      gives      0      cr
128    mv      gives      900    cr
256 mv gives 1800 cr
```

Программные «трюки»

Аппроксимация полиномами

```
cos                    f-f'                cr
hart                   3300                cr
-0.0043 0.0794 -0.6459 0.5708 indent
   2* 2* . triangle dup *. 2 poly indent
   -281 , 5203 , -42329 , 37407 , indent
   push drop pop *. + ;
```