

®

PIC24FJ256DA210

Графический модуль

2011

Александр Левчук

Оглавление

Оглавление	2
Введение.....	4
Функциональные возможности модуля GFX.....	4
Входные сигналы дисплейного контроллера	6
Выходные сигналы дисплейного контроллера	8
Адресация пикселей	9
Цветовая таблица (CLUT)	10
Отображение данных RGB на выходные контакты	12
Формирование изображения на TFT дисплее.....	13
Формирование синхронизирующих сигналов.....	17
Подключение дисплея TFT	20
Формирование изображения на дисплее STN	21
Режимы интерфейса STN и формирование синхронизирующих сигналов	24
Отображение данных на дисплеях STN.....	26
Поочередное переключение дисплейных данных.	27
Генератор тестового шаблона	28
Модули графической обработки (GPUs).....	29
Интерфейс команд	29
Состояние очереди команд FIFO	29
Модуль обработки графических символов (CHRGPU).....	31
Формат таблицы шрифтов.....	32
Команды CHRGPU	34
Операции CHRGPU.....	35
Модуль копирования прямоугольников (RCCGPU)	36
Команды RCCGPU	38
Модуль декомпрессии данных (IPU)	40
Команды IPU	40
Прерывания.....	41
Прерывания GPU	41
Прерывания буфера команд FIFO	41
Прерывания синхроимпульсов импульса	42
Последовательность подачи питания дисплея	42
Операции в режиме энергосбережения	42
Воздействие сигнала сброса	43
Регистры модуля	44
Общие управляющие регистры	44
Регистр команд GPU	44
Управляющие регистры дисплея G1CON1 - G1CON3	45
Регистр состояния G1STAT	48
Регистр разрешения прерываний G1IE	49
Регистр состояния прерываний G1IR.....	50

PIC24FJ256DA210. Графический модуль

Регистры рабочих областей и дисплейного буфера	51
Регистры стартового адреса рабочих областей G1W1ADR и G1W2ADR	51
Регистры ширины и высоты рабочей области G1PUW и G1PUH	52
Регистры стартового адреса дисплейного буфера G1DPADR	52
Регистры ширины и высоты дисплейного буфера G1DPW и G1DPH	52
Регистры общей ширины и высоты дисплея G1DPWT и G1DPHT	53
Регистры синхронизации	53
Регистр активной области дисплея G1ACTDA	53
Регистр горизонтальной синхронизации G1HSYNC	54
Регистр вертикальной синхронизации G1VSYNC	54
Регистр синхронизирующего импульса G1DBLCON	54
Регистр установки опережения прерывания G1MRGN	55
Регистры цветовой таблицы	55
Регистр записи данных в цветовую таблицу G1CLUTWR	55
Регистр чтения данных из цветовой таблицы G1CLUTRD	55
Управляющий регистр цветовой таблицы G1CLUT	56
Другие регистры	57
Регистры координат символа G1CHRX и G1CHRY	57
Регистр состояния модуля распаковки изображения G1IPU	57
Регистр разрешения контактов выходных данных G1DBEN	58
Сводная таблица регистров	59

Введение

Модуль графического контроллера (GFX) создан для прямого управления дисплеями со встроенным аналоговым контроллером попиксельного управления. Фактически микропроцессор содержит универсальный контроллер дисплея, который позволяет управлять дисплеями двух типов, TFT и разновидностями дисплеев типа STN, как цветными, так и монохромными. Совместно с хорошей программной поддержкой в виде бесплатной графической библиотеки и дисплеями с сенсорной панелью, микропроцессор с модулем GFX позволяет создавать более дешевые и гибкие устройства, чем, если бы использовать дисплей с собственным контроллером и отдельной клавиатурой. В тоже время наличие модуля GFX не отрицает использования в устройстве дисплея с собственным контроллером.

Модуль GFX обеспечивает ускоренное отображение вертикальных и горизонтальных линий, прямоугольников, копирование прямоугольных областей, рисование текста и декомпрессию сжатых данных. Ускоренное отображение выполняется за счет использования буфера команд FIFO. Однократная загрузка модуля видеоданными и командами позволяет в дальнейшем аппаратно обновлять изображение, освобождая CPU для других задач. Программа может опрашивать состояние модуля или использовать прерывания для последующего обновления экрана.

Наличие высокоуровневой графической библиотеки избавляет разработчика от рутинной работы по низкоуровневому программированию интерфейса пользователя и отображения графики.

Функциональные возможности модуля GFX

Модуль GFX поддерживает три типа дисплеев:

- Монохромные STN
- Цветные STN
- Цветные TFT

Модуль GFX позволяет программировать временную последовательность управляющих сигналов и данных для широкого спектра дисплеев. Он формирует временную диаграмму синхронизирующих сигналов на основе тактовой частоты GCLK, ширины и высоты дисплея в пикселях. Остальные параметры синхросигналов, такие как полярность, задержки и другие, программируются в регистрах модуля в драйвере дисплея.

Графическая библиотека содержит драйверы, поддерживающие основные модели дисплеев. Открытый код, наличие большого количества примеров и модульность драйвера позволяет легко написать собственный драйвер для отсутствующего в библиотеке типа дисплея или откорректировать существующий.

Модуль GFX имеет следующими особенностями:

- Содержит аппаратный графический ускоритель, который содержит:
 - модуля символьной графической обработки (CHRGPU)
 - модуля копирования прямоугольников (RCCGPU)
 - модуля распаковки сжатого изображения (IPU)
- Поддерживает цветовую таблицу (CLUT) на 256 цветов.
- Поддерживает цветовую глубину 1/2/4/8/16 бит на пиксель (bpp).
- Имеет программируемое разрешение дисплея.
- Поддерживает следующие дисплейные интерфейсы:
 - 4/8/16-битные монохромные дисплеи типа STN
 - 4/8/16-битные цветные дисплеи типа STN

PIC24FJ256DA210. Графический модуль

- 9/12/18/24- битные цветные дисплеи типа TFT (18 и 24-битные дисплеи подключаются как 16-битные цветового формата 565 RGB)

Модуль состоит (рисунок 1) из командного интерфейса, который принимает команды процессора посредством модуля регистров, трех модулей обработки(GPU), контроллера дисплея и арбитра памяти.

Модули обработки специализированы по выполняемым функциям:

- Модуль RCCGPU рисует прямоугольники с заливкой цветом. В подмножество прямоугольников входит точка, вертикальная и горизонтальная линия. Прямоугольники могут копироваться и модифицироваться, также возможны растровые операции и операции с прозрачными цветами.
- Модуль символьной обработки CHRGPU может отображать строки с выбором цвета фона и цвета переднего плана, а также управлять масками. Фон может быть прозрачным.
- Модуль декомпрессии IPU (Inflate Processing Unit) восстанавливает изображение предварительно сжатое по алгоритму сжатия данных без потерь (RFC 1951).

Дисплейный контроллер непрерывно обновляет изображение из дисплейного буфера, если GPU имеет доступ к памяти. Дисплейный буфер и рабочая область GPU могут быть отображены на один и тот же блок памяти. Только один модуль GPU может быть активным. Необходимости в написании специальных подпрограмм для управления модулями обработки нет, все они управляются командами, содержащимися в 16-ти уровне стеке FIFO. Число команд незначительно, их использование для языка C скрыто функциями графической библиотеки.

Частота обновления, цветовая глубина и разрешение используются для определения параметров контроллера дисплея. Контроллер дисплея поддерживает дисплеи TFT с параллельной шиной RGB, цветные и монохромные STN дисплеи. Для STN используется 16 уровневое сглаживание. Цветовая таблица доступна для обоих типов дисплеев.

В некоторых устройствах системная RAM может быть внешней. Внешняя системная RAM используется в основном для цветных дисплеев, когда недостаточно ресурсов внутренней памяти для хранения видеобуфера. Особенно потребность в ресурсах памяти возрастает при использовании двойной буферизации для устранения мерцания изображения. Для цветных дисплеев в большинстве случаев необходима внешняя память

Интерфейсный блок модуля GFX реализует временную диаграмму сигналов дисплея. Эти сигналы делятся на две группы, данные и сигналы синхронизации. Загрузка данных в дисплей производится в параллельно последовательном коде с шириной шины 4, 8 или 16 бит.

Сигналы синхронизации

Синхронизирующие сигналы содержат:

- Сигнал горизонтальной синхронизации HSYNC. Данный сигнал определяет начало каждой строки пикселей. По этому сигналу дисплей фиксирует данные одной строки в своей памяти.
- Сигналы тактовой частоты GCLK и разрешения данных GEN.
- Сигнал вертикальной синхронизации VSYNC определяет начало отображения, нулевую (первую) отображаемую строку.
- Сигнал GPWR включает дисплей. При использовании этого сигнала для управления источником питания дисплея формируется определенная последовательность подачи питания на дисплей. Сначала подается питание, а затем остальные сигналы, при выключении сначала снимаются сигналы модуля, а затем отключается питание.

PIC24FJ256DA210. Графический модуль

Все синхронизирующие сигналы формируются из частоты DISPCLK, которая формируется в синтезаторе частоты микропроцессора из частоты G1CLK. Синхронизирующие сигналы имеют программируемые параметры, длительность и задержку относительно начала временной диаграммы. Если используется дисплей, имеющий поддержку в графической библиотеке, то программисту можно не заботиться о программировании графического модуля и не вникать в тонкости аппаратной реализации дисплея. Библиотечный драйвер дисплея обеспечит его правильное функционирование. В противном случае следует тщательно изучить документацию на дисплей для написания собственного драйвера или корректировки наиболее подходящего библиотечного.

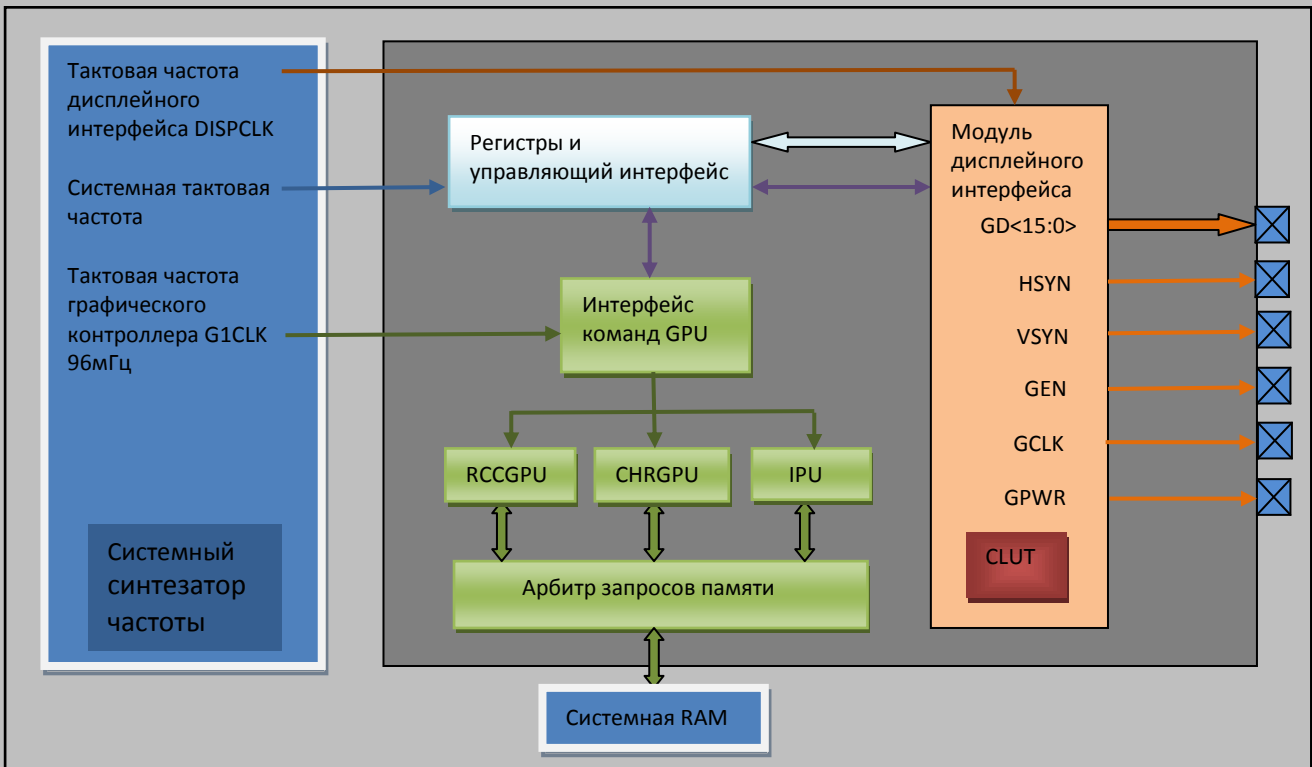


рисунок 1

Входные сигналы дисплейного контроллера

Дисплейный контроллер использует сигналы синтезатора частоты микропроцессора (рисунок 1) для формирования выходных сигналов. Временные диаграммы сигналов зависят от типа дисплея.

На рисунок 2 показана часть схемы синтезатора частоты, формирующая входные частоты для модуля GFX.

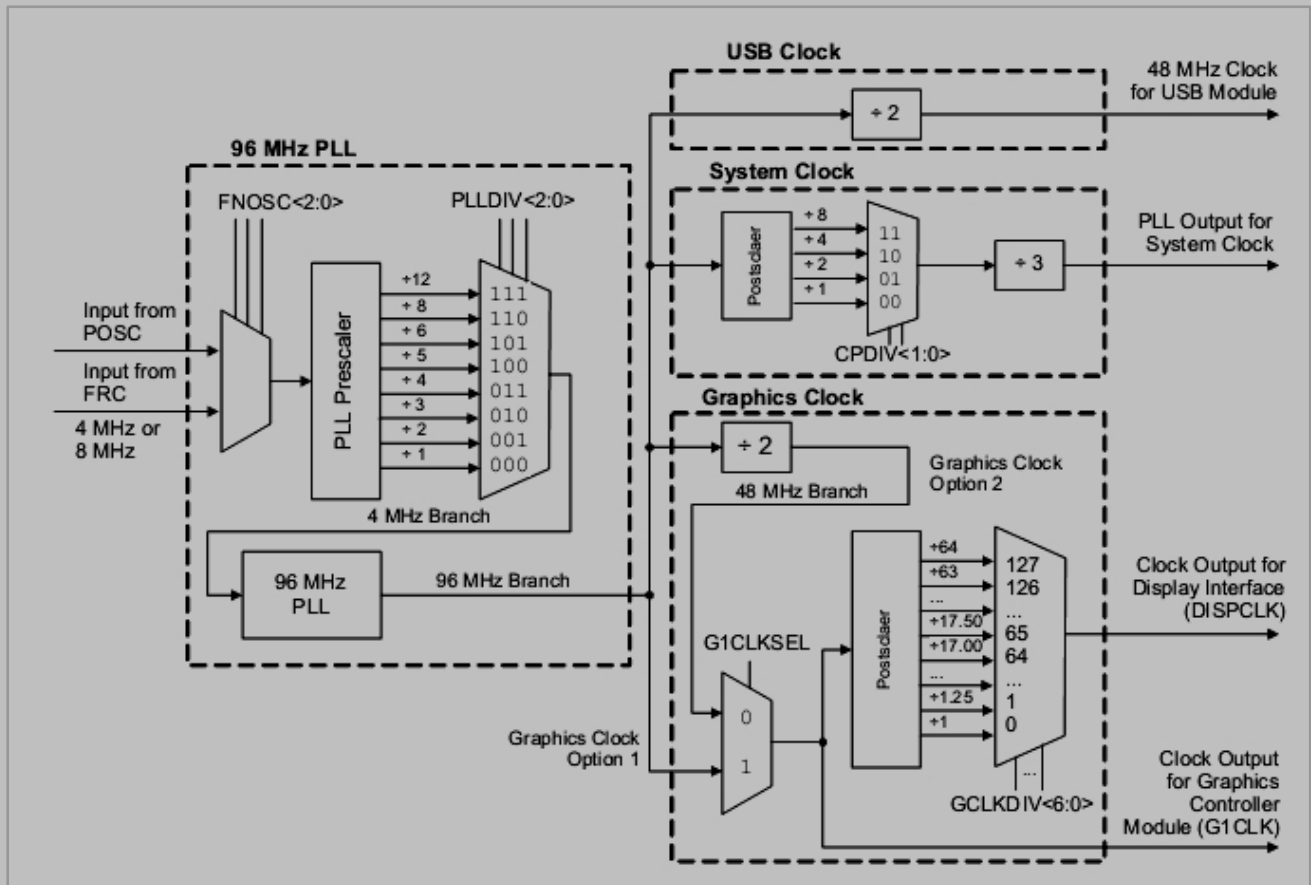


рисунок 2

- **G1CLK** – частота 96 или 48 мГц, формируемая синтезатором частоты с ФАПЧ(PLL) микропроцессора (биты G1CLKSEL).
- **System Clock** – системная частота микропроцессора от 4 до 32мГц. Формируется делителем из частоты 96мГц, поступающей от синтезатора частоты. Устанавливается битами CPDIV<1:0> в регистре CLKDIV.
- **DISPCLK** - выходной сигнал синтезатора частоты для дисплейного интерфейса. Формируется делителем из частоты **G1CLK**. Шаг перестройки частоты различный для разных коэффициентов деления. Коэффициент деления (K_D) устанавливается в битах GCLKDIV<6:0> (CLKDIV2<15:9>) . Определить частоту **DISPCLK** для входной частоты 96(48) мГц можно по формуле:

$$F = F_{96(48)} / (1 + n \times K_D),$$

где $n = 0.25$ для $K_D = 0 - 63$ или $n = 0.5$ для $K_D = 64 - 95$ или $n = 1$ для $K_D = 96 - 127$;

В [таблица 1](#) приводится пример расчета нескольких частот для разных n . Для каждого типа дисплеев частота DISPCLK указывается в его технических данных. При отсутствии документации на дисплей следует определить среднюю частоту для этого типа дисплеев (TFT, STN) и затем подобрать по наилучшему отображению на дисплее. Также можно найти коэффициент деления в исходных кодах драйверов дисплея графической библиотеки.

PIC24FJ256DA210. Графический модуль

таблица 1

Kd		n - цена мл. разряда	(1 + n × Kd) - делитель входной частоты	DISPCLK
GCLKDIV<6:0>	Десятичное значение			
0000000	0	0.25	1	96 MHz (48 MHz)
0000001	1	0.25	1.25	76.80 MHz (38.4 MHz)
0000010	2	0.25	1.5	64 MHz (32 MHz)
.....				
0111111	63	0.25	16.75	5.73 MHz (2.86 MHz)
1000000	64	0.25	17	5.65 MHz (2.82 MHz)
1000001	65	0.5	17.5	5.49 MHz (2.74 MHz)
1000010	66	0.5	18	5.33 MHz (2.66 MHz)
... ..				
1011111	95	0.5	32.5	2.95 MHz (1.47 MHz)
1100000	96	1	33	2.91 MHz (1.45 MHz)
1100001	97	1	34	2.82 MHz (1.41 MHz)
1100010	98	1	35	2.74 MHz (1.37 MHz)
.....				
1111110	126	1	63	1.52 MHz (762 kHz)
1111111	127	1	64	1.50 MHz (750 kHz)

Выходные сигналы дисплейного контроллера

Дисплейный контроллер (рисунки 1) формирует несколько выходных синхронизирующих сигналов и данные, которые выдаются на порты микропроцессора:

- **GCLK** – выходной порт графического модуля для тактовой частоты сдвига данных. Частота этого сигнала определяется автоматически в соответствии с разрядностью шины данных и типом дисплея. Частота GCLK для дисплеев TFT всегда равна частоте DISPCLK. Для дисплеев MSTN/CSTN частота GCLK зависит от ширины шины данных. Период GCLK для этих дисплеев определяется следующим образом:
 - Для CSTN 4-бита, $T = \text{DISPCLK} \times 2$.
 - Для MSTN 4-бита/CSTN 8-бит, $T = \text{DISPCLK} \times 4$.
 - Для MSTN 8-бит/CSTN 16-бит, $T = \text{DISPCLK} \times 8$.
 - Для MSTN 16-бит, $T = \text{DISPCLK} \times 16$.
 Значение частоты **GCLK** определяется по данным документации на дисплей, где указывается предельно допустимая тактовая частота. Затем производится расчет и установка частоты **DISPCLK** в регистре CLKDIV2.
- **GEN** – порт графического модуля для частоты разрешающего сигнала, которая равна частоте сигнала **GCLK**, но сдвинута по фазе на 180°. Дополнительно может сдвигаться установкой бит HENST<7:0> в регистре G1DBLCON. Этот сигнал изначально трактуется как сигнал разрешения загрузки данных, но может использоваться и как тактовый сигнал вместо GCLK. Его применение определяется характеристиками дисплея.
- **HSYNC** – порт горизонтальной синхронизации дисплея. Сигнал используется для фиксации данных одной строки в памяти дисплея.
- **VSNC** – порт вертикальной синхронизации дисплея. Сигнал определяет начало отображения, первую (нулевую) отображаемую строку.
- **GPWR** – порт включения питания дисплея.

Адресация пикселей

В зависимости от цветовой глубины каждое слово памяти (16 бит) содержит разное число пикселей. При глубине цвета в 1 бит слово памяти содержит 16 пикселей, соответственно при глубине цвета 16 бит слово памяти содержит 1 пиксель. В результате, в зависимости от цветовой глубины, один и тот же пиксель на дисплее будет размещаться по разным адресам в дисплейном буфере.

GPU при записи в дисплейный буфер использует цветовую глубину, установленную в регистре PUBPP<2:0> (G1CON1<7:5>), а при чтении буфера используется цветовая глубина, считанная из регистра DPBPP<2:0> (G1CON2<7:5>). Значения в этих регистрах должны быть эквивалентны для обеспечения правильного отображения.

Некоторые приложения обрабатывают изображения в отдельной памяти с разной цветовой глубиной. Программа, в этом случае, должна устанавливать DPBPP и переключать дисплейный буфер по адресу изображения или копировать изображение в дисплейный буфер.

Существует два способа копирования изображения в дисплейный буфер, используя GPU или программно. В любом случае следует определять адреса пикселей в дисплейном буфере по их координатам на дисплее. Вычисление основывается на цветовой глубине и "шаге памяти" дисплейного буфера. Шаг памяти это смещение, в байтах добавляемое к адресу пикселя для определения адреса следующего пикселя следующей строки в той же самой колонке дисплейного буфера.

$$\text{Шаг памяти} = (\text{Ширина дисплея} * \text{цветовую глубину}) / 8.$$

Примечание.

Модуль графического дисплея поддерживает ширину дисплея кратную 16 пикселям.

Для MSTN дисплеев минимальная ширина 32 пикселя, для дисплеев с 16-битным интерфейсом минимальная ширина 48 пикселей.

	Address	0	1	2	3	4	...	59	60	61	62	63
0	DPADR<23:0>+0	P0	P1	P2	P3	P4	...	P59	P60	P61	P62	P63
1	DPADR<23:0>+8	P64	P65	P66	P67	P68	...	P123	P124	P125	P126	P127
2	DPADR<23:0>+16	P128	P129	P130	P131	P132	...	P187	P188	P189	P190	P191
...	...	...	...	...	...	...	...	...	...	...	...	...
5	DPADR<23:0>+40	P320	P321	P322	P323	P324	...	P379	P380	P381	P382	P383
6	DPADR<23:0>+48	P384	P385	P386	P387	P388	...	P443	P444	P445	P446	P447
7	DPADR<23:0>+56	P448	P449	P450	P451	P452	...	P507	P508	P509	P510	P511
8	DPADR<23:0>+64	P512	P513	P514	P515	P516	...	P571	P572	P573	P574	P575
9	DPADR<23:0>+72	P576	P577	P578	P579	P580	...	P635	P636	P637	P638	P639
10	DPADR<23:0>+80	P640	P641	P642	P643	P644	...	P699	P670	P671	P672	P673
...	...	...	...	...	...	...	...	...	...	...	...	...
22	DPADR<23:0>+176	P1408	P1409	P1410	P1411	P1412	...	P1467	P1468	P1469	P1470	P1471
23	DPADR<23:0>+184	P1472	P1473	P1474	P1475	P1476	...	P1531	P1532	P1533	P1534	P1535

рисунок 3

На [рисунок 3](#) показан пример для дисплея 64x24, с дисплейным буфером, начинающимся с адреса DPADR<23:0>.

PIC24FJ256DA210. Графический модуль

Шаг памяти вычисляется как: $(64 * 1) / 8 = 8$. Если изображение 59x17 (красный цвет) копируется в дисплейный буфер в расположение 2,6 или P386, смещение адреса пикселя вычисляется как:

$$\text{Смещение адреса пикселя} = (\text{Шаг памяти} * Y \text{ позицию}) + (X \text{ позиция} * \text{цветовую глубину} / 8)$$

Подставив результат $(8 * 6) + (2 / 8)$, получим смещение 48, игнорируя остаток от деления. Остаток определяет позицию бита пикселя P386 в байте памяти. Таким образом, адрес байта левого верхнего пикселя изображения равен DPADR<23:0> + 48.

Общая формула вычисления адреса пикселя для дисплея с размером $m \times n$ и m кратным 16:

$$\text{Смещение адреса пикселя} = \text{Относительный адрес пикселя} * \text{цветовая глубина} / 8$$

Вычисление относительного адреса байта пикселя показано в [таблица 2](#).

таблица 2

0	0	1	2	3	...	m-3	m-2	m-1
1	m	m + 1	m + 2	m + 3	...	m + (m - 3)	m + (m - 2)	m + (m - 1)
2	m*2	(m*2) + 1	(m*2) + 2	(m*2) + 3	...	(m*2) + (m - 3)	(m*2) + (m - 2)	(m*2) + (m - 1)
3	m*3	(m*3) + 1	(m*3) + 2	(m*3) + 3	...	(m*3) + (m - 3)	(m*3) + (m - 2)	(m*3) + (m - 1)
...	...	...	...	...	...	...	...	...
n-3	m*(n-3)	(m*(n-3)) + 1	(m*(n-3)) + 2	(m*(n-3)) + 3	...	(m*(n-3)) + (m - 3)	(m*(n-3)) + (m - 2)	(m*(n-3)) + (m - 1)
n-2	m*(n-2)	(m*(n-2)) + 1	(m*(n-2)) + 2	(m*(n-2)) + 3	...	(m*(n-2)) + (m - 3)	(m*(n-2)) + (m - 2)	(m*(n-2)) + (m - 1)
n-1	m*(n-1)	(m*(n-1)) + 1	(m*(n-1)) + 2	(m*(n-1)) + 3	...	(m*(n-1)) + (m - 3)	(m*(n-1)) + (m - 2)	(m*(n-1)) + (m - 1)

Например, используется дисплей с $m = 320$ и $n = 240$. Левый верхний угол изображения имеет относительные координаты $X = 3$ и $Y = 3$. Вычисляя, мы получим $(320 * 3) + 3 = 963$.

Для цветовой глубины 4bpp, пиксель располагается по адресу: DPADR + $(963 * 4 / 8)$ или DPADR + 481. Остаток, который определяет положение пикселя в адресуемом байте, игнорируется.

Цветовая таблица (CLUT)

Контроллер дисплея может использовать цветовую таблицу (256 элементов) для представления цвета пикселей в дисплейном буфере. Если использование таблицы разрешено, то подразумевается, что каждый пиксель в дисплейном буфере имеет цветовой индекс. Цветовой индекс является адресом в таблице, где содержится цвет данного пикселя. Цветовая таблица не отображается на системную память, доступ производится только через регистры SFR.

Размер цветовой таблицы и число ее элементов зависит от цветовой глубины ([таблица 3](#)).

таблица 3

Цветовая глубина DPBPP	Размер таблицы CLUT
1	2
2	4
4	16
8	256
16	256

При использовании 16-ти разрядных данных для каждого пикселя цветовой индекс содержится в младшем байте, старший байт игнорируется. В целом это приводит к общему увеличению памяти без улучшения качества изображения. Поэтому рекомендуется запрещать цветовую таблицу при работе с 16-ти разрядными данными.

Для дисплеев TFT и CSTN каждая запись таблицы является 16-ти разрядным цветом, не зависимо от установки цветовой глубины или

разрядности дисплейного интерфейса ([таблица 4](#)).

Для дисплеев MSTN каждая запись таблицы является 4-х разрядной градацией серого (16 полос).

таблица 4

Тип выхода	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGB	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
Яркость	Не используется												M3	M2	M1	M0

Выходы цветовой таблицы CLUT отражаются на дисплейные контакты, зависящие от типа дисплея, регистр DPMODE (G1CON2<2:0>), и от разрядности дисплейных данных, регистр DPGWDTH (G1CON2<15:14>). Данные цветовой таблицы CLUT не инициализируются по сигналу сброса Reset и не гарантируется их предыдущее значение после сигнала сброса. Приложение должно программировать таблицу перед первым использованием.

Цветовая таблица большую часть времени занята дисплейным контроллером в режиме обновления изображения и для обеспечения доступа к ней со стороны процессора необходимо иметь определенные временные промежутки. В режиме TFT доступ к таблице может быть организован во время вертикального или горизонтального синхронизирующего импульса.

В режиме STN обновление изображения происходит постоянно и получить доступ к таблице значительно сложнее. Microchip рекомендует, во избежание мерцания экрана, при доступе к таблице со стороны процессора выключать дисплей. Однако, данный способ весьма сомнителен, дисплей STN медленный и избежать мерцания дисплея таким способом вряд ли удастся.

Для чтения или записи таблицы должна быть проведена следующая подготовка.

Запись в CLUT:

1. Установить биты адреса CLUTADR (G1CLUT<7:0>).
2. Установить бит записи CLUTRWEN (G1CLUT<8>).
3. Записать данные в регистр G1CLUTWR.
4. Прочитать бит состояния CLUTBUSY (G1CLUT<14 и проверить его. Если 0, данные были записаны, если "1" ждать, пока не очистится.
5. Для следующей записи перейти на пункт 1.
6. После записи всех данных очистить бит CLUTRWEN.

Чтение CLUT:

1. Установить биты адреса CLUTADR.
2. Установить бит CLUTRWEN.
3. Установить бит CLUTTRD для получения данных из CLUT.
4. Очистить бит CLUTTRD.
5. Читать бит состояния CLUTBUSY с проверкой, если '0' - данные можно читать, если '1' - ждать очистки бита состояния.
6. Читать данные из регистра G1CLUTRD.
7. Для следующего чтения перейти на пункт 1.
8. После чтения всех данных очистить CLUTRWEN.

Очистка CLUTRWEN обеспечивает доступ дисплейного контроллера к CLUT.

Dithering (смешивание, размытие, сглаживание)

Сглаживание всегда разрешено для режимов MSTN и CSTN и не используется для режима TFT.

Примечание.

Термин dithering, который уже можно встретить в русской транскрипции как "дитеринг", не что иное, как цифровой фильтр, который осуществляет сглаживание исходной функции, применительно к видео или звуковым данным, способом размывания погрешности конечного отсчета.

PIC24FJ256DA210. Графический модуль

При использовании фильтра Dithering для видеоданных, формируются матрицы из нескольких пикселей. Внутри матрицы межпиксельная погрешность сглаживается, добавлением к соседним пикселям уменьшенной на определенную долю погрешности. Например, 7/16 погрешности правому пикселю, 3/16 нижнему левому пикселю. В результате резкие переходы изображения сглаживаются.

Отображение данных RGB на выходные контакты

Для режима TFT, без использования цветовой таблицы (CLUT), шина данных показана в [таблица 5](#). Если используется CLUT, отображение данных на контакты будет таким же, только теперь цветные данные будут определяться индексом таблицы. Для дисплеев CSTN и MSTN, отображение данных на контакты зависит от разрядности шины и организации шинного интерфейса.

таблица 5

Color	Memory Value	Red<4:0> GD<15:11>	Green<5:0> GD<10:5>	Blue<4:0> GD<4:0>
1 bpp				
Black	0	00000	000000	00000
White	1	11111	111111	11111
2 bpp				
Black	00	00000	000000	00000
Dark Gray	01	01000	010000	01000
Light Gray	10	11000	110000	11000
White	11	11111	111111	11111
4 bpp				
Black	0000	00000	000000	00000
Dark Blue	0001	00000	000000	10000
Dark Red	0010	10000	000000	00000
Dark Magenta	0011	10000	000000	10000
Dark Green	0100	00000	100000	00000
Dark Cyan	0101	00000	100000	10000
Dark Yellow	0110	10000	100000	00000
Gray	0111	10000	100000	10000
Black	1000	00000	000000	00000
Blue	1001	00000	000000	11111
Red	1010	11111	000000	00000
Magenta	1011	11111	000000	11111
Green	1100	00000	111111	00000
Cyan	1101	00000	111111	11111
Yellow	1110	11111	111111	00000
White	1111	11111	111111	11111
8 bpp				
	D<7:0>	D<7,6,5,5,5>	D<4,3,2,2,2,2>	D<1,0,0,0,0>
16 bpp				
	D<15:0>	D<15:11>	D<10:5>	D<4:0>

Формирование изображения на TFT дисплее.

В графических дисплеях пиксели расположены в координатной сетке X,Y. Для уменьшения объема оборудования обычно используется мультиплексный доступ к пикселям. На координату X (по строкам) производится поочередная подача напряжения на сегменты выбранных пикселей, а на координату Y (столбцы) поочередно подается напряжение на общий электрод (рисунок 4). В результате "светится" тот пиксель, на который подано сегментное напряжение и одновременно на общий электрод подан потенциал для подсветки пикселя. Для обеспечения синхронизации мультиплексного режима отображения и периодической загрузки новых данных дисплей имеет внутреннюю память, модуль синхронизации и интерфейсный модуль. Сегментный и общий драйверы обеспечивают преобразование цифровых сигналов в аналоговые, которые, в конечном счете, управляют жидкими кристаллами. Таков общий принцип создания изображения на графическом дисплее. По аналогии с формированием развертки в аналоговых электронно-лучевых трубках синхронизирующие сигналы делятся на сигналы горизонтальной развертки или строчной и сигналы вертикальной развертки или кадровой. В действительности их функция совершенно иная, синхросигналы обеспечивают загрузку данных в память дисплея, в отличие от аналогового дисплея.

Сигналы горизонтальной развертки (HSYNC) обеспечивают формирование сегментных данных (по числу пикселей) из поступающих входных данных. Входные данные имеют небольшую разрядность, от последовательного кода различных протоколов до параллельно-последовательного кода. Разрядность параллельно-последовательного кода соответствует числам 4, 8, 16. Входные данные заполняют сдвигающий регистр и затем переписываются в память.

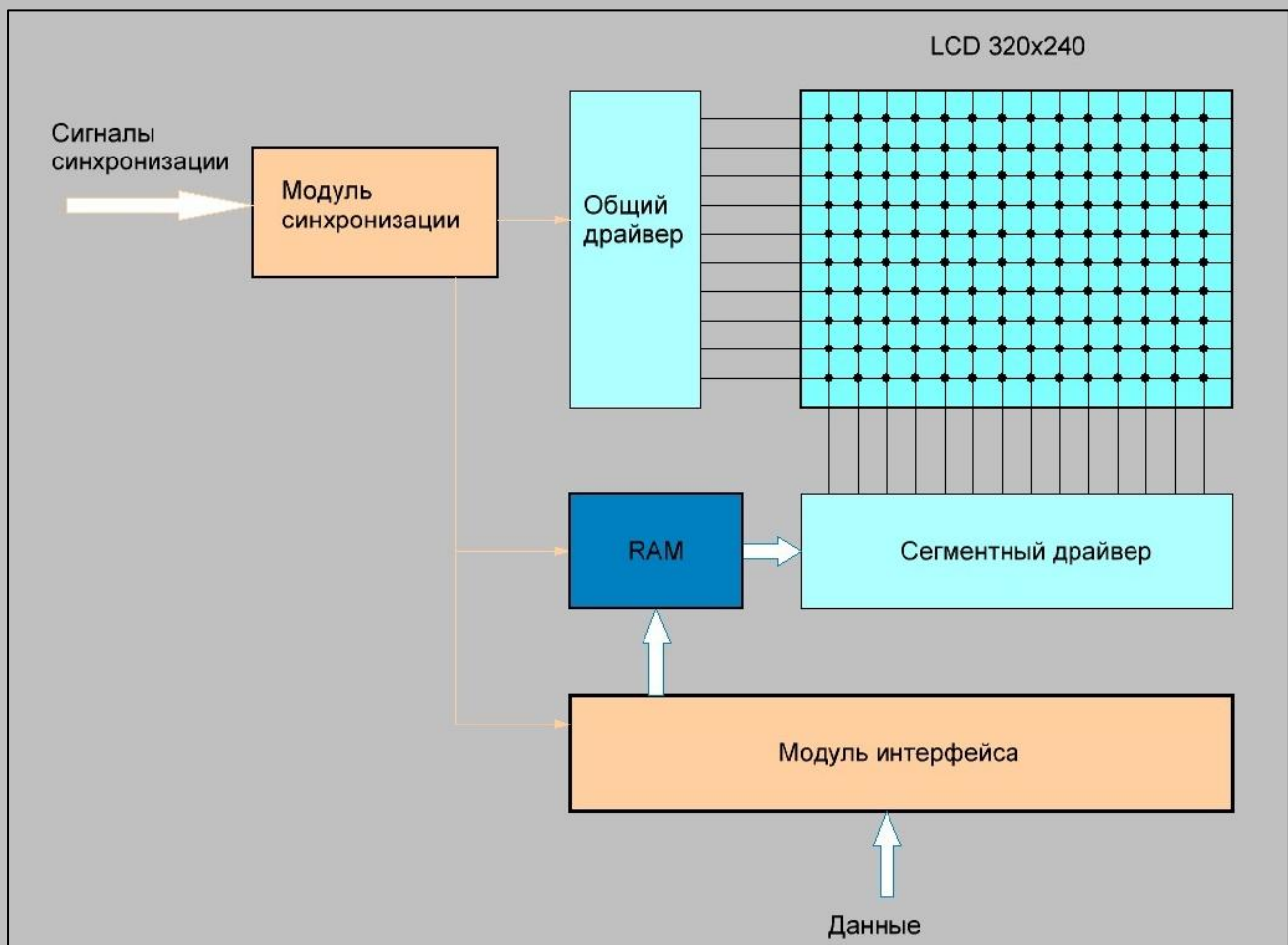


рисунок 4

PIC24FJ256DA210. Графический модуль

Сигналы вертикальной развертки (VSYNC) синхронизируют счетчик строк, обеспечивая правильное положение первой и последующих строк изображения. Структура поступающих данных должна соответствовать RGB данным. Соответственно и этот интерфейс называется RGB интерфейс.

В зависимости от состава оборудования различают дисплеи с контроллерами и без контроллеров. Такое деление на категории не совсем правомерно, любой современный дисплей имеет, по крайней мере, хотя бы один контроллер - аналоговый контроллер управления пикселями. В абсолютном большинстве случаев дисплеи имеют несколько контроллеров. Когда говорят о дисплее с контроллером, то подразумевают наличие дополнительного контроллера, который обеспечивает цифровой интерфейс приема данных (часто стандартный) и самостоятельно вырабатывает синхросигналы для остальной части дисплея. Дисплей без контроллера требует подачи синхросигналов горизонтальной и вертикальной развертки, а также данных RGB. Такое деление можно считать достаточно условным, многие дисплеи имеют несколько различных интерфейсов и обеспечивают работу как с интерфейсом RGB, так и, например, с интерфейсами 6800, 8080, SPI.

Изображение на дисплее TFT создается в центральной части экрана, которая называется активной областью. По периметру экрана создаются небольшие неактивные (защитные) области. Размер защитной области по вертикали составляет несколько строк, а по горизонтали несколько десятков пикселей. Для организации защитных промежутков на синхронизирующих сигналах выделяются соответствующие области (рисунок 5). В таблице 6 показаны ограничения на параметры синхронизации и формулы для их расчета.

Формирование временной диаграммы синхросигналов HSYNC, VSYNC, GEN, GCLK производится программированием ряда управляющих регистров модуля GFX. Программированию подлежат биты управляющих регистров, показанные в таблице 7, где для примера, указаны параметры типового дисплея портретной конфигурации со следующими техническими параметрами:

- Тактовая частота GCLK – 183 ns
- Горизонтальный период HSYNC – 280 периодов GCLK
- Передний горизонтальный защитный промежуток – 10 периодов GCLK
- Задний горизонтальный защитный промежуток – 20 периодов GCLK
- Импульс HSYNC – 10 периодов GCLK
- Вертикальный период – 326 строк
- Передний вертикальный защитный промежуток – 2 строка
- Задний вертикальный защитный промежуток – 1 строка
- Импульс VSYNC – 3 строки

Данные для программируемых параметров определяются из технической документации на дисплей. Вся временная диаграмма сигналов сжимается или растягивается в зависимости от выбора коэффициента деления K_d , который определяет частоту DISPCLK.

В свою очередь DISPCLK выбирается исходя из технических параметров дисплея, с некоторым запасом. При испытаниях желательно сначала выбрать среднее значение частоты DISPCLK, например, для дисплея Truly TFT-G240320LTSW-118W-E частота составляет около 6 мГц. В дальнейшем все ссылки на реальный дисплей будут подразумевать именно TFT-G240320LTSW-118W-E.

[illegible][illegible]

PIC24FJ256DA210. Графический модуль

Длительность строки	—	DPW	—	$T_{GCLK} = T_{DISPCLK}$
Время до первого активного пикселя	1	HENST ($T_6 + T_7 + T_9$)	—	$T_{GCLK} = T_{DISPCLK}$

таблица 7

Параметр	Регистр	Биты регистра	Значение	Описание
Общие параметры дисплея				
Тип дисплея	G1CON2	DPMODE	1	TFT дисплей.
Ширина дисплея	G1DPW	DPW	240	Активная ширина.
Высота дисплея	G1DPH	DPH	320	Активная высота.
Общая ширина дисплея	G1DPWT	DPWT	280	Указывается в спецификации дисплея.
Общая высота дисплея	G1DPHT	DPHT	326	Указывается в спецификации дисплея.
Параметры шины данных				
Разрядность шины данных	G1CON2	DPGWDTH	X	Игнорируется. Для TFT дисплеев всегда 16 бит.
Разрешение выходных контактов данных	G1DBEN	GDBEN	0xFFFF	Разрешены все 16 контактов данных.
Разрешение сигнала разрешения данных	G1CON3	DPENOE	1	Сигнал разрешения дисплейных данных (GEN) разрешен.
Активный уровень сигнала разрешения данных	G1CON3	DPENPOL	1	Высокий.
Фиксация данных по фронту/спаду	G1CON3	DPCLKPOL	0	Фиксация данных по спаду тактового сигнала.
Параметры вертикальной синхронизации				
Разрешение VSYNC	G1CON3	DPVSOE	1	Сигнал VSYNC разрешен.
Активный уровень VSYNC	G1CON3	DPVSPOL	0	Активный уровень VSYNC - низкий.
Задержка до начала VSYNC	G1VSYNC	VSST	2	Равняется переднему защитному промежутку.
Длительность импульса VSYNC	G1VSYNC	VSLEN	3	Указывается в спецификации дисплея.
Начало вертикальной развертки	G1DBLCON	VENST	6	Число строк от начала вертикальной развертки до конца импульса VSYNC.
Активная строка	G1ACTDA	ACTLINE	6	Число строк перед первой активной строкой.
Параметры горизонтальной развертки				
Активный уровень HSYNC	G1CON3	DPHSPOL	0	Низкий.
Разрешение HSYNC	G1CON3	DPHSOE	1	Разрешает использовать HSYNC.
HSYNC Start	G1HSYNC	HSST	10	Равняется переднему защитному промежутку.
HSYNC Length	G1HSYNC	HSLEN	10	Указывается в спецификации дисплея.
Horizontal Enable Start	G1DBLCON	HENST	40	Число пикселей до первого активного.
Активный пиксель	G1ACTDA	ACTPIX	40	Число пикселей до первого активного.
Общие разрешающие биты				
Разрешение дисплейных выходов	G1CON3	DPPINOE	1	Выходы разрешены.
Управление сигналом включения питания дисплея	G1CON3	DPPOWER	1	Включить питание дисплея.
Разрешение сигнала питания дисплея	G1CON3	DPPWROE	1	Разрешен.
Разрешение модуля дисплейного контроллера	G1CON1	G1EN	1	Дисплейный модуль разрешен.

Формирование синхронизирующих сигналов

Формирование сигналов вертикальной синхронизации.

Импульс вертикальной синхронизации вместе с защитными промежутками, по аналогии с аналоговой разверткой, называется бланкирующим промежутком или импульсом обратного хода развертки. Из [таблица 7](#) видно, что для создания этого сигнала нужно запрограммировать регистры G1VSYNC(биты VSST, VSLEN), G1DBLCON, G1ACTDA микропроцессора. Не следует забывать, что первый регистр, который нужно запрограммировать для создания временной диаграммы это регистр синтезатора частоты CLKDIV2. Исходя из [рисунок 5](#) и [таблица 7](#), можно нарисовать отдельно временную диаграмму сигнала вертикальной синхронизации с защитными промежутками и программируемыми параметрами([рисунок 6](#)).

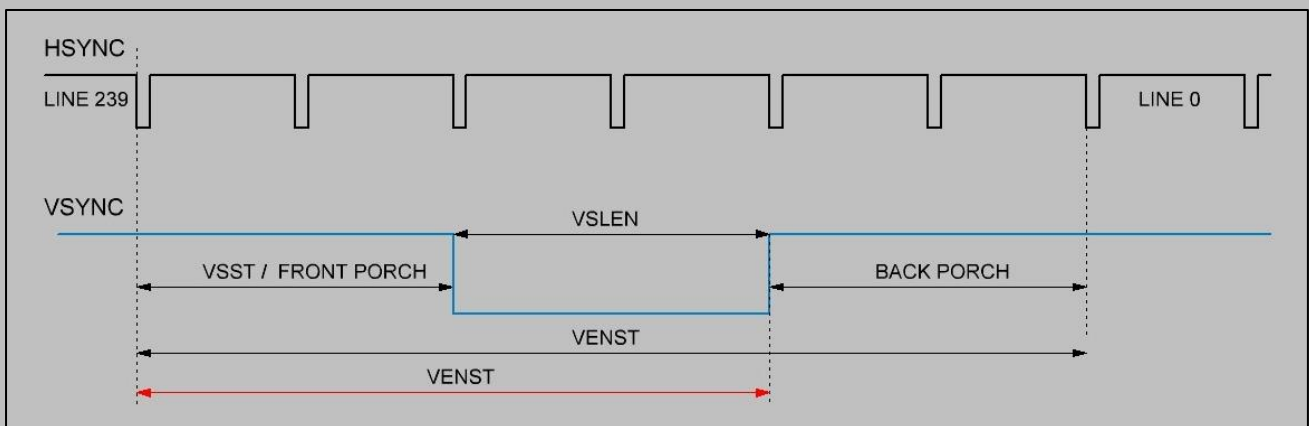


рисунок 6

Для формирования переднего защитного промежутка используются биты VSST. Как мы видим, для заднего защитного промежутка нет программируемых битов ни в одном регистре, он просто задается для вычисления остальных программируемых параметров. Заднего защитного промежутка может и не быть, поэтому биты VENST показаны на рисунке в двух вариантах. В описании графического модуля микропроцессора о битах VENST дословно сказано следующее: " This is the number of lines from the start of vertical blanking to the first displayed line of a frame". Что в переводе означает: "Это число строк от начала бланкирующего промежутка до первой отображаемой строки в кадре".

Также существует регистр первой активной (видимой) строки G1ACTDA, в котором программируются биты ACTLINE. Назначение этих битов, точно такое же, как и битов VENST. В упоминаемом документе Microchip на этот счет сказано следующее: " Typically, ACTLINE = VENST (G1DBLCON<15:8>). This register is added for versatility in the timing of the active line. For TFT mode the minimum value is 2. For STN mode the minimum value is '0'. " В переводе это означает, что регистр добавлен для большей гибкости управления первой видимой строкой, с предупреждением, что минимальное значение регистра равно 2 для режима TFT и ноль для режима STN. Обычно значение этого параметра равно параметру VENST.

Проведем проверочную настройку дисплея Truly TFT-G240320LTSW-118W-E в составе отладочной платы Microchip.

Если следовать документации и запрограммировать параметры в соответствии с рекомендациями и руководствуясь примером из [таблица 7](#), то на реальном изображении появляются артефакты в виде дополнительной полосы приблизительно в две строки. В документации на дисплеи обычно не приводится никаких данных по его подключению и формированию сигналов, информацию по этим параметрам можно найти только в документации на сегментный драйвер дисплея, в данном случае драйвер SSD1289. Документ позволяет скопировать временную диаграмму SSD1289, которая показана ниже на [рисунок 7](#).

PIC24FJ256DA210. Графический модуль

Эта временная диаграмма соответствует рекомендациям Microchip и примеру в [таблица 7](#), но не соответствует реальной картинке, видимой на экране. Для прояснения ситуации обратимся к программному драйверу дисплея из состава графической библиотеки Microchip, который формирует изображение на экране правильно.

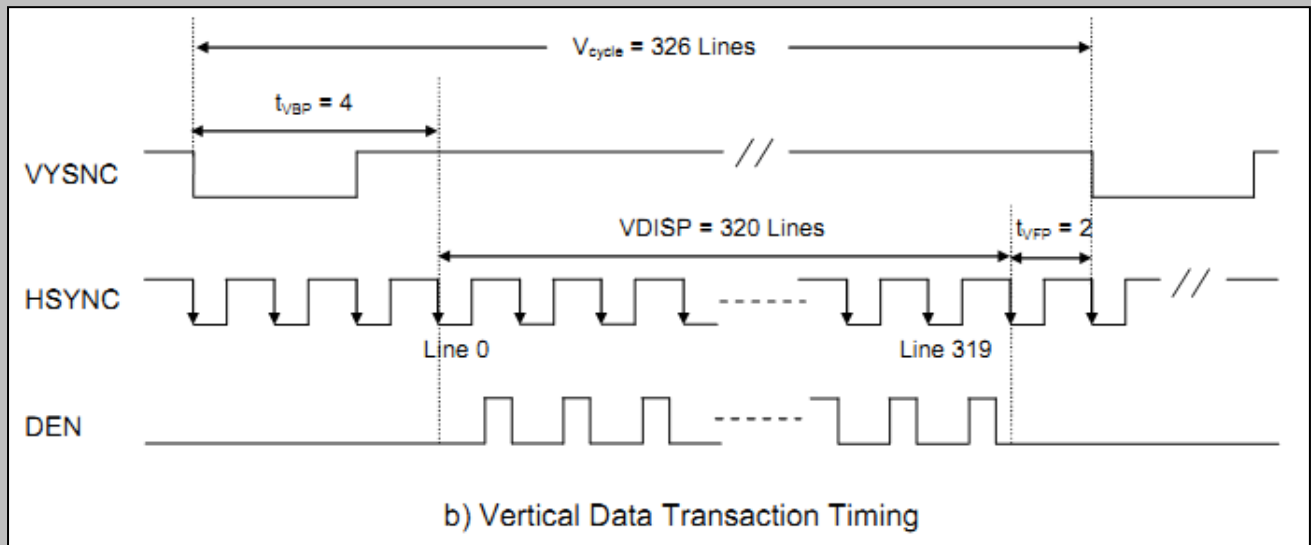


рисунок 7

Драйвер, как оказалось, пренебрегает рекомендациями и устанавливает параметры следующим образом:

- $VSST = 0$, передний защитный промежуток.
- $VSLEN = 4$, длительность импульса синхронизации.
- $VENST = 4$, что не соответствует описанию параметра в документации Microchip, где указана формула для его вычисления $VENST = VSST + VSLEN + \text{Vertical Back Porch} = 0 + 4 + 2 = 6$.
- $ACTLINE = 6$, время до первой активной строки.

Следует обратить внимание на тот факт, что параметр $VENST$ не является определяющим для времени до начала активной строки. Определяющим параметром является $ACTLINE$, а параметр $VENST$ определяет, по моему мнению, значение $VSST + VSLEN$, в данном случае.

Таким образом, рекомендации и формулы вычисления констант в документации Microchip и SSD1289 оказались весьма условными и требующими коррекции. Для данного случая задний защитный промежуток определяется как $ACTLINE - VENST$ (или $VSLEN$). Две величины остаются достоверными и требуют точного соответствия, это общая высота дисплея $DPHT = 326$ пикселей и высота активной области $DPH = 320$ пикселей. Для правильного определения общего, не активного промежутка вертикальной развертки вычисляем $ACTLINE = DPHT - DPH$, затем программируем остальные величины в соответствии с документом на сегментный драйвер, и глядя на изображение, корректируем параметры $VSST$, $VSLEN$ и $VENST$. Наибольшее влияние оказывает передний защитный промежуток $VSST$.

На практике следует опираться на данные, приводимые в документации на дисплей, точнее на документацию не самого дисплея, а документацию на контроллер дисплея. К сожалению, она не всегда доступна. Очевидно, что именно для таких случаев в составе графического модуля есть тестовый генератор, который не требует наличия драйвера дисплея, достаточно запрограммировать регистры в соответствии с [таблица 7](#) и произвести отладку на тестовом сигнале. Затем следует внести правильные константы в драйвер дисплея.

Формирование сигналов горизонтальной синхронизации.

Для горизонтальной синхронизации формируется сигнал HSYNC и сигнал GEN. Сигнал GEN равен по длительности активной части строки и разрешает считывание данных строки. Сигнал HSYNC имеет защитные промежутки, исчисляемые в периодах частоты $GCLK = DISPCLK$. Расчет программируемых значений ведется, так же как и для вертикальной развертки. На копии рисунка (рисунок 8) из документации сегментного драйвера SSD1289 показаны сигналы горизонтальной синхронизации и их параметры.

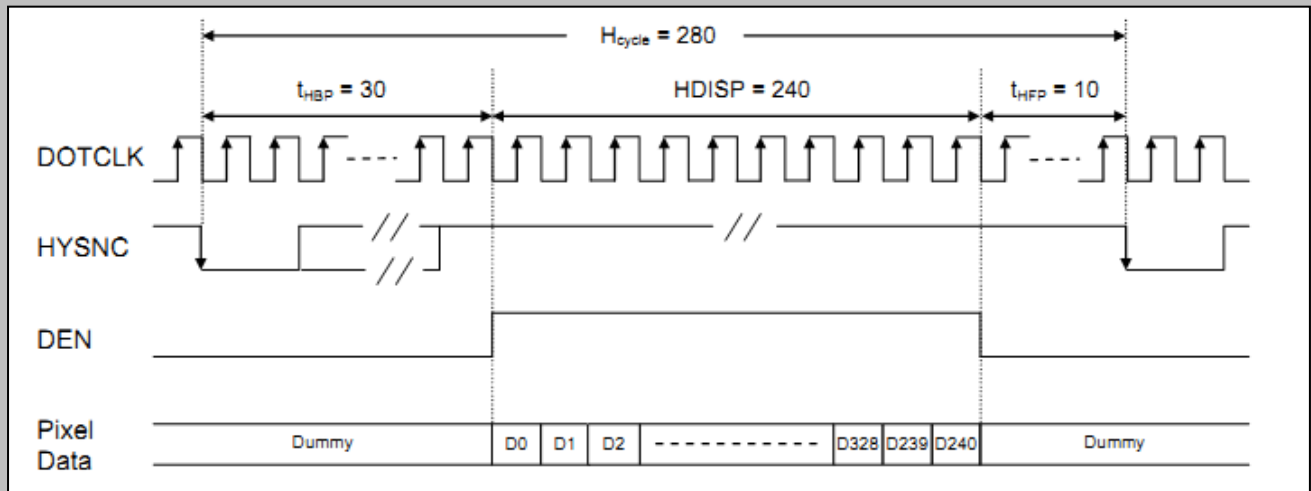


рисунок 8

Основой расчета программируемых параметров является длительность не активной части строки $ASTPIX = DPWT - DPW$, затем, используя данные из документации на сегментный драйвер, программируем остальные значения. В документации на сегментный драйвер SSD1289 приводится сводная таблица (рисунок 9) рекомендуемых параметров.

Characteristics	Symbol	Min	Typ	Max	Unit
DOTCLK Frequency	f_{DOTCLK}	-	5.5	8.22	MHz
DOTCLK Period	t_{DOTCLK}	122	182	-	nSec
Horizontal Frequency (Line)	f_H	-	19.6	29.3	kHz
Vertical Frequency (Refresh)	f_V	-	60	90	Hz
Horizontal Back Porch	t_{HBP}	-	30	-	t_{DOTCLK}
Horizontal Front Porch	t_{HFP}	-	10	-	t_{DOTCLK}
Horizontal Data Start Point	t_{HBP}	-	30	-	t_{DOTCLK}
Horizontal Blanking Period	$t_{HBP} + t_{HFP}$	-	40	-	t_{DOTCLK}
Horizontal Display Area	HDISP	-	240	-	t_{DOTCLK}
Horizontal Cycle	H_{cycle}	-	280	-	t_{DOTCLK}
Vertical Back Porch	t_{VBP}	-	4	-	Line
Vertical Front Porch	t_{VFP}	-	2	-	Line
Vertical Data Start Point	t_{VBP}	-	4	-	Line
Vertical Blanking Period	$t_{VBP} + t_{VFP}$	-	6	-	Line
Vertical Display Area	VDISP	-	320	-	Line
Vertical Cycle	V_{cycle}	-	326	-	Line

рисунок 9

Сравнивая данные таблица 9 и данные рисунок 8, с реальными параметрами программного драйвера отмечаем еще худшее совпадение в настройке параметров горизонтальной синхронизации по сравнению с параметрами вертикальной синхронизации. Программный драйвер графической библиотеки Microchip устанавливает следующие значения параметров:

- HSLen = 25, длительность импульса;

PIC24FJ256DA210. Графический модуль

- HSST = 0, передний защитный промежуток;
- АСТPIХ = HENST = 30, время до начала активного пикселя;

Задний защитный промежуток равен АСТPIХ – HSLen = 5. Замечаем, что длительность импульса и задний порог отдельно не указываются, а регламентируется их сумма. Как видим, не соблюдается даже время до первого активного пикселя, не активный промежуток короче на 10 пикселей, при сохранении общего размера в 280 пикселей. Десять пикселей исчезли куда-то, но изображение правильное. Попробуем проверить установки программного драйвера дисплея. Установим передний защитный промежуток в соответствии с документацией SSD1289, HSST = 10 и посмотрим на результат. Изображение осталось правильное, ничего не изменилось. Можно предположить, что программист, по аналогии с вертикальной синхронизацией, установил нулевое значение и, увидев правильную картинку, оставил все как есть. Однако, в процессе работы с данным дисплеем, иногда наблюдался устойчивый сбой строчной синхронизации, возможно, он был вызван именно этой причиной, поэтому исправление в драйвере оставляем для дальнейшего наблюдения. Если сбой синхронизации больше не будет проявляться, тогда это такая коварная ошибка в коде драйвера.

По результатам практической настройки можно сделать вывод, что для настройки дисплея следует хорошо понимать принцип формирования изображения на дисплее TFT и первоначально руководствоваться значениями параметров из документации на сегментный драйвер дисплея. Затем, воспользовавшись тестовым генератором микропроцессора, настроить правильное изображение. В дальнейшем эти параметры устанавливаются в новом драйвере дисплея.

Подключение дисплея TFT

Интерфейс RGB графического модуля выводит данные в виде 16-битного кода в формате RGB 5-6-5. Для согласования с дисплеями TFT которые требуют данных в формате 18-бит (6-6-6) RGB или стандартном 24-bit (8-8-8) RGB используется простая аппаратная схема перекодировки.

таблица 8

Graphics Data Pins (GDx)	18-Bit RGB Display (6-6-6)	24-Bit RGB Display (8-8-8)
RED		
GD<15:11>	Red<5:1>	Red<7:3>
GD<15>	Red<0>	Red<2:0>
GREEN		
GD<10:5>	Green<5:0>	Green<7:2>
GD<10>	Green<5>	Green<1:0>
BLUE		
GD<4:0>	Blue<5:1>	Blue<7:3>
GD<4>	Blue<0>	Blue<2:0>

Для этого все биты данных модуля переносятся в старшие разряды данных дисплея. Свободные младшие разряды дисплея (3-2-3 для 24-битных дисплеев, 1-0-1 для 18-битных) подключаются к старшим разрядам модуля. Это позволяет устанавливать полную яркость, когда старший разряд равен 1, и устанавливать полностью черное

поле, когда старший разряд равен 0. Подробности показаны в [таблица 8](#).

Современные дисплеи TFT с *интерфейсом RGB* имеют сложные сегментные контроллеры, которые требуют первоначальной настройки внутренних цепей с использованием собственных команд программирования внутренних регистров. Программирование сегментного драйвера производится через его системный интерфейс. Сегментные драйверы могут иметь несколько системных интерфейсов как параллельных, так и последовательных. Чаще всего для этих целей используется интерфейс SPI. Не следует путать указанные дисплеи с дисплеями, которые полностью управляются по интерфейсу SPI или I²C. Первоначальная инициализация дисплея производится специальным программным драйвером, который может поставляться с графической библиотекой Microchip для

определенной номенклатуры сегментных драйверов. Если таковой отсутствует в библиотеке, то следует запросить код инициализации сегментного драйвера у производителя дисплея. Некоторые производители предоставляют код инициализации, но весьма высока вероятность его недоступности. В таком случае придется изучать внутреннюю структуру сегментного драйвера, его систему команд и заниматься экспериментированием, что чревато потерей времени и не гарантирует положительного результата на приемлемом отрезке времени. Поэтому перед началом разработки необходимо внимательно отнестись к выбору дисплея не только по внешним параметрам, но и учесть доступность программной поддержки самого дисплея. Лучшим вариантом является выбор дисплея из состава предлагаемых библиотекой Microchip, всего около 20 различных типов дисплеев разных производителей с программной и схемной поддержкой.

Формирование изображения на дисплее STN

Общий принцип формирования изображения на дисплеях STN и улучшенных разновидностях этой технологии (FSTN и другие) не отличается от дисплеев TFT. Отличия заключаются в более простом формировании синхросигналов и существовании шин данных уменьшенной разрядности. Данные передаются по шине, разрядность которой представляется рядом 4,8,16. Дисплеи STN бывают монохромные (MSTN) и цветные (CSTN). Синхросигналы для обеих групп дисплеев формируются без защитных промежутков. В дисплеях STN нет неактивных строк и неактивных пикселей.

Интерфейс дисплеев STN может функционировать в трех режимах, режим сдвига по тактовой частоте (Shift Clock), переключения по кадру (Frame Toggle) и переключения по строкам (Line Toggle). Чаще всего используется режим сдвига по тактовой частоте, остальные два режима требуют дополнительного сигнального входа MOD. Управление режимом производится битом DPENPOL. Если бит установлен, то модуль формирует сигналы для режима Shift Clock, сброшенный бит является признаком функционирования в одном из двух оставшихся режимов. При переключении в режим Frame Toggle или Line Toggle изменяется тип сигнала на выходе GEN. Для режима Shift Clock, сигнал GEN по частоте соответствует инвертированному сигналу GCLK и может дополнительно сдвигаться относительно GCLK. Сдвиг измеряется в периодах DISPCLK. Для режима Frame Toggle сигнал GEN является меандром с периодом равным VSYNC при условии установки битов VENST = 0. Для режима Line Toggle, при значении VENST = N, сигнал GEN имеет длительность равную $T_{H\text{SYNC}} * N$.

Сигнал GCLK для дисплеев MSTN/CSTN имеет период равный DISPCLK умноженный на константу, определяемую разрядностью шины данных:

- Для CSTN 4-бита, $T = \text{DISPCLK} * 2$.
- Для MSTN 4-бита/CSTN 8-бит, $T = \text{DISPCLK} * 4$.
- Для MSTN 8-бит/CSTN 16-бит, $T = \text{DISPCLK} * 8$.
- Для MSTN 16-бит, $T = \text{DISPCLK} * 16$.

В [таблица 9](#) приводятся примерные параметры регистров для дисплея MSTN со следующими типовыми данными:

- Период тактовой частоты GCLK – 71 ns (minimum)
- Горизонтальный период – 80 периодов GCLK
- Горизонтальный синхроимпульс – 1 период GCLK
- Вертикальный период – 240 строк
- Вертикальный синхроимпульс – 1 строка

В [таблица 10](#) приведен пример конфигурации дисплейного контроллера для цветного дисплея CSTN со следующими типовыми данными:

- Период тактовой частоты GCLK – 25 ns (minimum)
- Горизонтальный период – 120 периодов GCLK

PIC24FJ256DA210. Графический модуль

- Горизонтальный синхроимпульс – 25 ns (minimum)
- Вертикальный период – 240 строк
- Вертикальный синхроимпульс – 1 строка

таблица 9

Параметр	Регистр	Биты регистра	Значение	Описание
Общие параметры дисплея				
Тип дисплея	G1CON2	DPMODE	2	MSTN дисплей.
Ширина дисплея	G1DPW	DPW	320	Ширина.
Высота дисплея	G1DPH	DPH	240	Высота.
Общая ширина дисплея	G1DPWT	DPWT	320	Равна ширине
Общая высота дисплея	G1DPHT	DPHT	240	Равна высоте.
Параметры шины данных				
Разрядность шины данных	G1CON2	DPGWDTH	0	Разрядность шины данных 4 бита.
Разрешение выходных контактов данных	G1DBEN	GDBEN	0x000F	Разрешены 4 мл. бита интерфейса данных.
Разрешение сигнала разрешения данных	G1CON3	DPENOE	1	Сигнал разрешения дисплейных данных (GEN) разрешен.
Активный уровень сигнала разрешения данных	G1CON3	DPENPOL	1	Высокий.
Фиксация данных по фронту/спаду	G1CON3	DPCLKPOL	0	Фиксация данных по спаду тактового сигнала.
Параметры вертикальной синхронизации				
Разрешение VSYNC	G1CON3	DPVSOE	1	Сигнал VSYNC разрешен.
Активный уровень VSYNC	G1CON3	DPVSPOL	1	Активный уровень VSYNC - высокий.
Задержка до начала VSYNC	G1VSYNC	VSST	0	-
Длительность импульса VSYNC	G1VSYNC	VSLEN	1	Указывается в спецификации дисплея.
Начало вертикальной развертки	G1DBLCON	VENST	x	-
Активная строка	G1ACTDA	ACTLINE	0	-
Параметры горизонтальной развертки				
Активный уровень HSYNC	G1CON3	DPHSPOL	1	Низкий.
Разрешение HSYNC	G1CON3	DPHSOE	1	Разрешает использовать HSYNC.
HSYNC Start	G1HSYNC	HSST	1	Задержка строчного импульса.
HSYNC Length	G1 HSYNC	HSLEN	4	Обычно равно ACTPIX.
Horizontal Enable Start	G1DBLCON	HENST	0	Задержки shift нет.
Активный пиксель	G1ACTDA	ACTPIX	4	Число периодов DISPCLK в периоде GCLK.
Общие разрешающие биты				
Разрешение дисплейных выходов	G1CON3	DPPINOE	1	Выходы разрешены.
Управление сигналом включения питания дисплея	G1CON3	DPPOWER	1	Включить питание дисплея.
Разрешение сигнала питания дисплея	G1CON3	DPPWROE	1	Разрешен.
Разрешение модуля дисплейного контроллера	G1CON1	G1EN	1	Дисплейный модуль разрешен.

PIC24FJ256DA210. Графический модуль

таблица 10

Параметр	Регистр	Биты регистра	Значение	Описание
Общие параметры дисплея				
Тип дисплея	G1CON2	DPMODE	3	CSTN дисплей.
Ширина дисплея	G1DPW	DPW	320	Ширина.
Высота дисплея	G1DPH	DPH	240	Высота.
Общая ширина дисплея	G1DPWT	DPWT	480	Вычисляемая величина: (DPW) * (3/2). В этом случае, $320 * 3/2 = 480$.
Общая высота дисплея	G1DPHT	DPHT	240	Равна высоте.
Параметры шины данных				
Разрядность шины данных	G1CON2	DPGWDTH	1	Разрядность шины данных 8 бита.
Разрешение выходных контактов данных	G1DBEN	GDBEN	0x00FF	Разрешены 8 мл. бита интерфейса данных.
Разрешение сигнала разрешения данных	G1CON3	DPENOE	1	Сигнал разрешения дисплейных данных (GEN) разрешен.
Активный уровень сигнала разрешения данных	G1CON3	DPENPOL	1	Высокий.
Фиксация данных по фронту/спаду	G1CON3	DPCLKPOL	0	Фиксация данных по спаду тактового сигнала.
Параметры вертикальной синхронизации				
Разрешение VSYNC	G1CON3	DPVSOE	1	Сигнал VSYNC разрешен.
Активный уровень VSYNC	G1CON3	DPVSPOL	1	Активный уровень VSYNC - высокий.
Задержка до начала VSYNC	G1VSYNC	VSST	0	-
Длительность импульса VSYNC	G1VSYNC	VSLEN	1	Указывается в спецификации дисплея.
Начало вертикальной развертки	G1DBLCON	VENST	x	-
Активная строка	G1ACTDA	ACTLINE	0	-
Параметры горизонтальной развертки				
Активный уровень HSYNC	G1CON3	DPHSPOL	1	Низкий.
Разрешение HSYNC	G1CON3	DPHSOE	1	Разрешает использовать HSYNC.
HSYNC Start	G1HSYNC	HSST	1	Задержка строчного импульса.
HSYNC Length	G1 HSYNC	HSLEN	4	Обычно равно ACTPIX.
Horizontal Enable Start	G1DBLCON	HENST	0	Задержки shift нет.
Активный пиксель	G1ACTDA	ACTPIX	4	Для CSTN величина вычисляется: $DPGWDTH (4, 8 \text{ or } 16)/2$. Таким образом величина равна 2, 4 and 8.
Общие разрешающие биты				
Разрешение дисплейных выходов	G1CON3	DPPINOE	1	Выходы разрешены.
Управление сигналом включения питания дисплея	G1CON3	DPPOWER	1	Включить питание дисплея.
Разрешение сигнала питания дисплея	G1CON3	DPPWROE	1	Разрешен.
Разрешение модуля дисплейного контроллера	G1CON1	G1EN	1	Дисплейный модуль разрешен.

1. Режим сдвига тактовой частоты (Shift Clock Mode)

Этот режим используется для дисплеев, которые получают данные, продвигая их во входном регистре, а затем защелкивают. Установка режима производится установкой бита $VENST = 1$. Поскольку входные схемы интерфейса могут быть различными, то в качестве частот сдвига графический модуль предлагает две тактовые частоты GCLK и GEN. Частота GEN предлагается для выполнения сдвига, как более универсальная в настройке, а GCLK для защелкивания данных. Не всегда требуется две частоты, есть дисплеи, использующие только одну тактовую частоту, а защелкивание данных они производят для полностью загруженной строки по сигналу HSYNC. Так работает, например, дисплей фирмы Volymín BP320240A. Для этого дисплея в качестве тактовой достаточно частоты GCLK, GEN можно не использовать и освободить контакт для других целей или наоборот, освободить контакт GCLK.

На [рисунок 10](#), в качестве примера, показана временная диаграмма загрузки данных для дисплея Volymín BP320240A. Эта временная диаграмма и последующие построены на основе осциллографических измерений и могут не совпадать в некоторых деталях с рисунками в документации Microchip. Упомянутый дисплей работает только в режиме Shift Clock. Для сдвига данных используется частота GEN, биты $HENST = 1$.

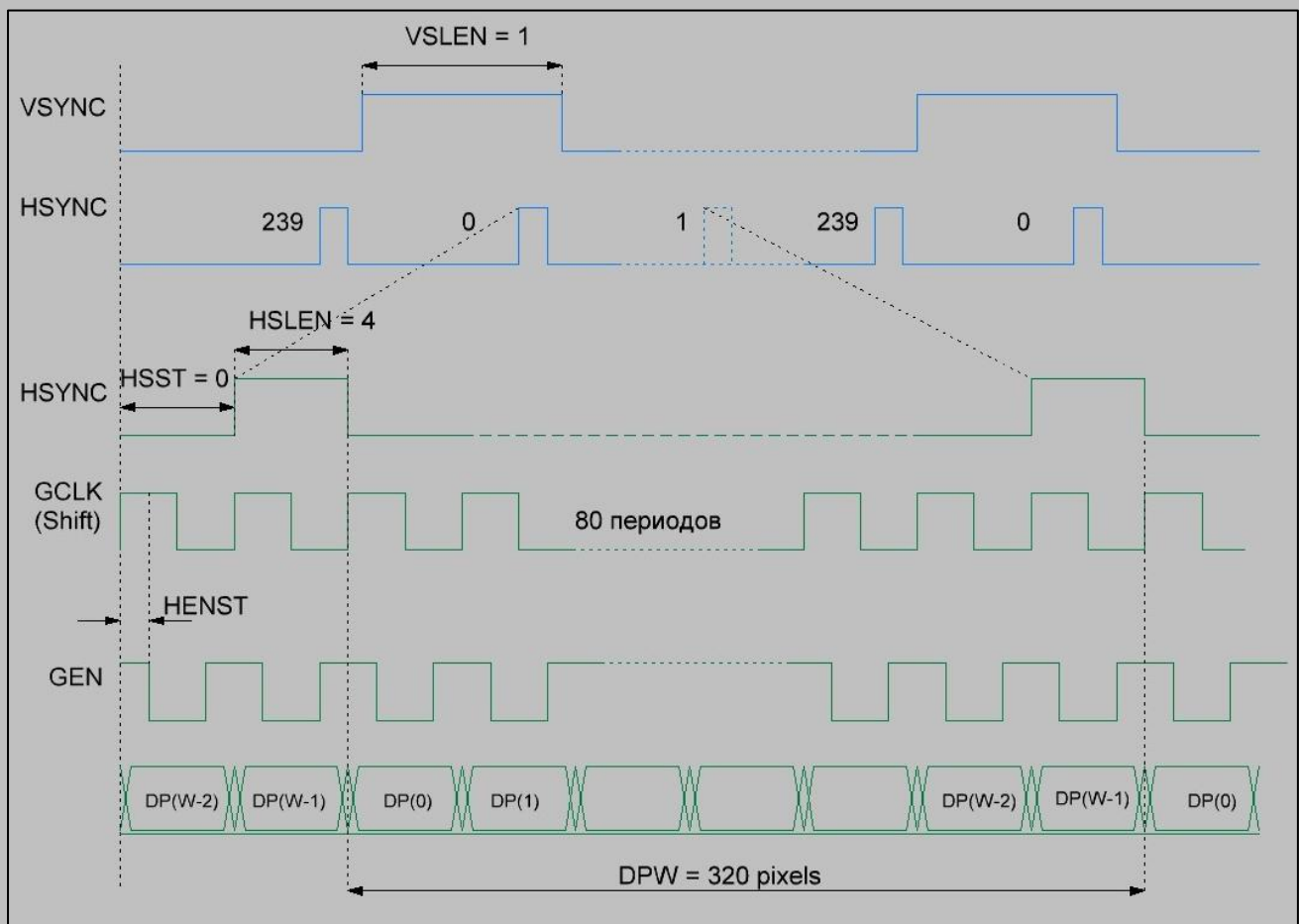


рисунок 10

Для проведения корректной замены сигнала GEN на GCLK следует сдвинуть HSYNS на один период DISPCCLK ($HSST = 1$), тогда взаимное положение сигналов HSYNS и GCLK становится таким же, как и для случая HSYNS и GEN при $HSST = 0$. Обратите внимание, что импульс VSYNC начинается с задержкой

PIC24FJ256DA210. Графический модуль

после 239 строки при HSST = 0. При периоде GCLK = 0,4 мксек задержка составляет около 2 мксек (~20 периодов DISPCLK). При HSST >> 0, 239-й импульс HSYNC начинает сдвигаться вправо, входя в область действия импульса VSYNC, на изображении при этом появляются искажения, поскольку счетчик строк очищается раньше фиксации данных последней строки.

2. Режим переключения кадров

В этом режиме на дополнительный вход MOD дисплея подается сигнал кадровой частоты, в виде меандра (рисунок 11), который формируется на выходе GEN.

Режим устанавливается в следующих регистрах:

- DPENPOL = 0
- VENST = 0
- HENST = не используется.

Сигналы дисплея подключаются к следующим выходам:

- GEN = MOD
- GCLK = тактовая частота.

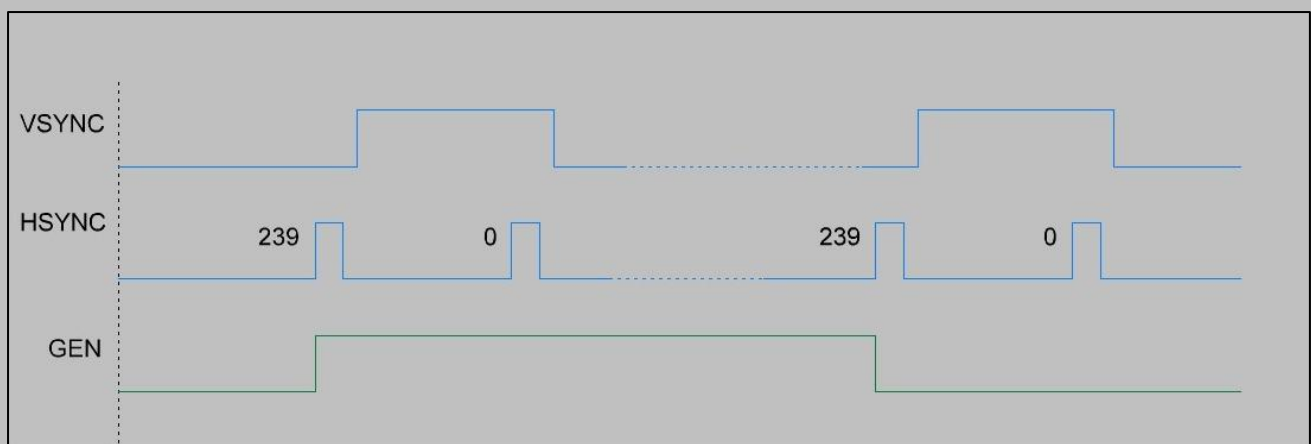


рисунок 11

3. Режим переключения строк

В этом режиме на вход MOD дисплея подается сигнал строчной частоты в виде меандра (рисунок 12). Частота программируется в параметре VENST > 0. Период сигнала GEN определяется по формуле $T_{\text{GEN}} = T_{\text{HSYNC}} * 2 * \text{VENST}$, где T_{HSYNC} период одной строки. Точные параметры сигнала можно определить из документации на дисплей, работающий в таком режиме. Дополнительно сигнал GEN может быть задержан N периодов частоты DISPCLK.

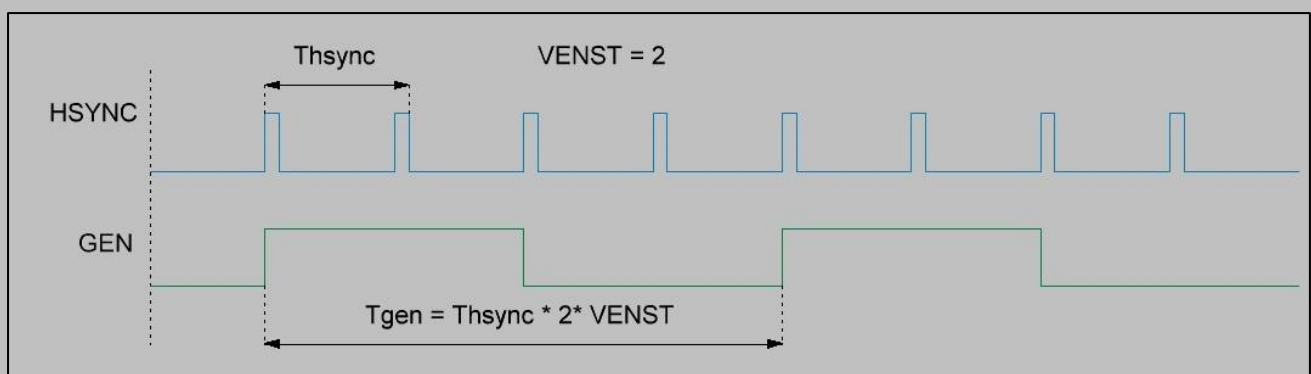


рисунок 12

Режим устанавливается в следующих регистрах:

PIC24FJ256DA210. Графический модуль

- DPENPOL = 0
- VENST = число, определяющее полупериод сигнала GEN.
- HENST = задержка сдвинутой тактовой частоты в периодах DISPCLK.

Сигналы дисплея подключаются к следующим выходам:

- GEN = MOD
- GCLK = тактовая частота.

Отображение данных на дисплеях STN

На рисунке рисунок 13 и рисунок 14 показано отображение пиксельных данных для монохромных (MSTN) и цветных (CSTN) дисплеев.

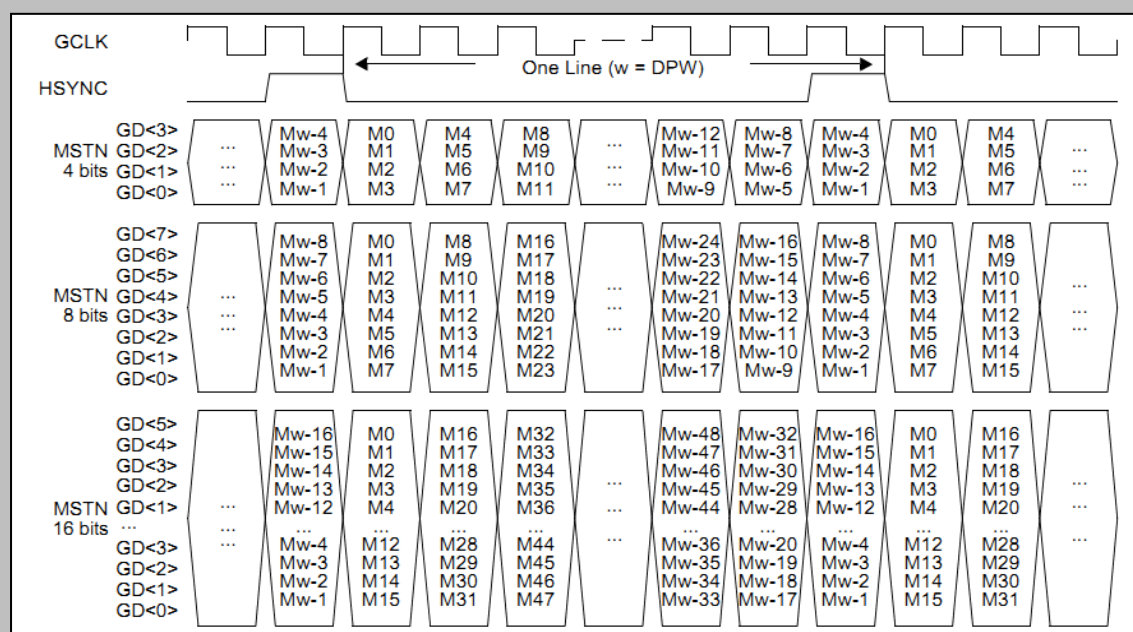


рисунок 13

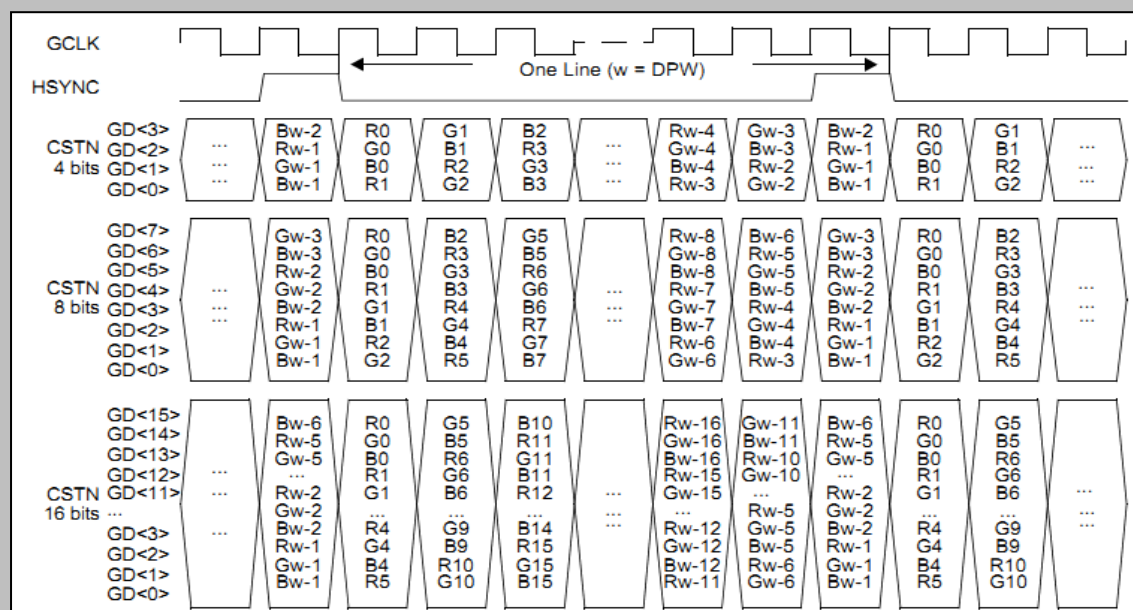


рисунок 14

Поочередное переключение дисплейных данных.

Чтобы уменьшить вероятность появления шума на земляной шине, вызываемый одновременным переключением сигналов данных, можно использовать метод поочередного переключения данных. Для этого существует четыре схемы поочередного переключения данных, которые определяются в битах DPSTGER(G1CON2<13:12>).

Когда DPSTGER установлен по умолчанию в 0, переключение данных синхронизировано по фронту GCLK. Когда DPSTGER = 0x01, переключение нечетных бит задерживается на $\frac{1}{2}$ периода GPUCLK. Когда DPSTGER = 0x10, переключение четных бит задерживается на один период GPUCLK.

Когда DPSTGER = 0x11, переключение сигналов организуется в четырех группах:

- GD<0>, GD<4>, GD<8>, GD<12> – без задержки
- GD<1>, GD<5>, GD<9>, GD<13> – задержка на $\frac{1}{2}$ GPUCLK
- GD<2>, GD<6>, GD<10>, GD<14> – задержка на 1 GPUCLK
- GD<3>, GD<7>, GD<11>, GD<15> – задержка на $1\frac{1}{2}$ GPUCLK

Малое время задержки, не оказывает никакого влияния на загрузку данных. Визуально, на дисплее сложно заметить результат работы данного режима, с помощью простого осциллографа также не удалось заметить изменения шумов на земляной шине.

Генератор тестового шаблона

Генератор тестового шаблона включен в графический модуль для тестирования дисплейного интерфейса. Генератор можно использовать для просмотра тестового шаблона, установив режим с помощью битов DPTEST<1:0> (G1CON2<9:8>). Интерфейс памяти при этом не используется, только формируется сетка синхросигналов и данные используемого шаблона. Применение генератора очень полезно для первоначальной оценки работоспособности дисплея с выбранными параметрами регистров. На [рисунок 15](#) показаны примеры четырех шаблонов.

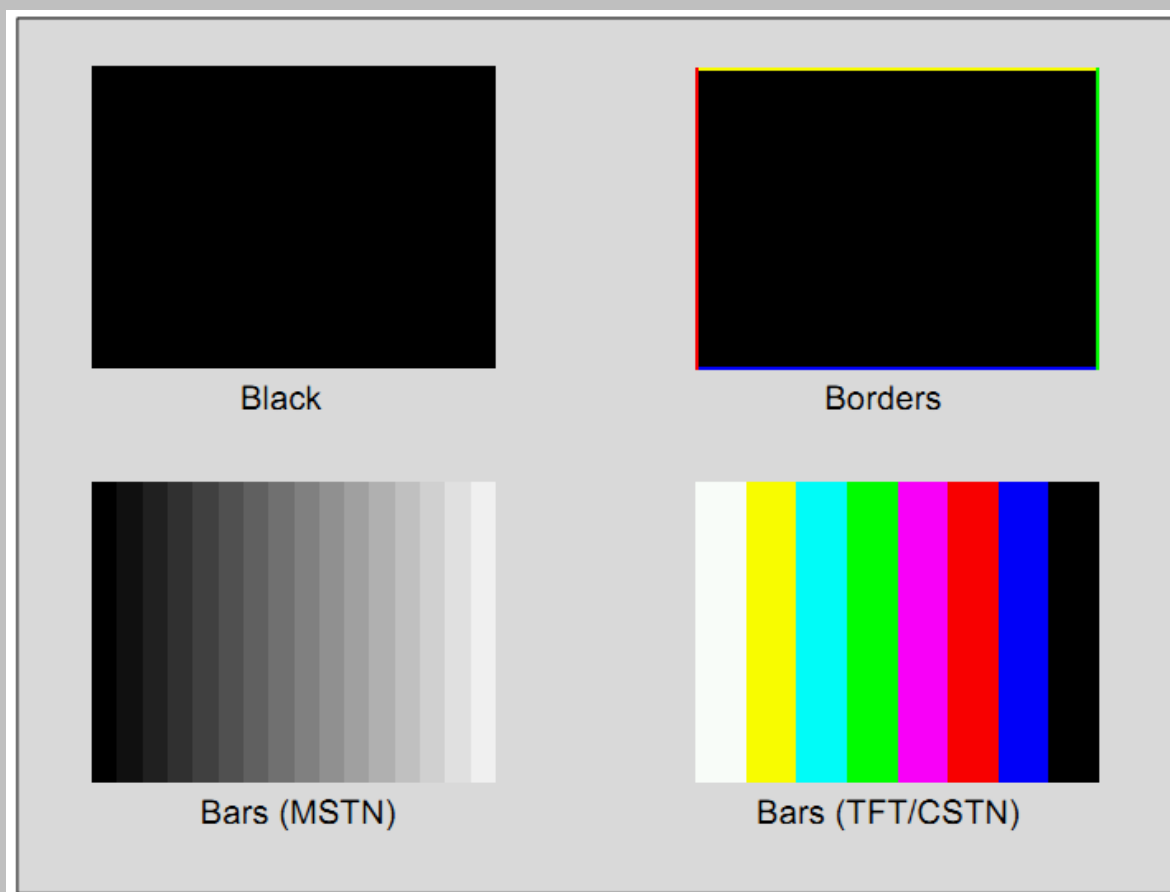


рисунок 15

Модули графической обработки (GPUs)

Интерфейс команд

Модули графической обработки управляются 32-х разрядными командами, которые записываются в регистр команд GCMD. Команда, записанная в регистр, пересылается затем в буфер команд FIFO глубиной в 16 команд. Команды проверяются и отправляются на исполнение в один из трех модулей GPU. Модули GPU могут работать только последовательно. Все команды в очереди FIFO будут ожидать завершения работы модуля GPU. Буфер очереди команд имеет программируемые условия своего состояния. Каждый модуль имеет свои специфические команды.

Формат команд:

- GCMD<31:28> 4 старших бита определяют адрес GPU.
- GCMD<27:24> Код команды.
- GCMD<23:0> Атрибуты команды.

Состояние очереди команд FIFO

Состояние очереди команд может быть проверено с помощью трех бит состояния регистра G1STAT.

- Бит опустошения CMDMPT (G1STAT<0>) указывает, что буфер пуст, но это не означает, что все GPU простаивают. Возможно, что последняя команда выполняется одним из GPU, в этом случае следует проверить состояние каждого GPU.
- Бит заполнения FIFO CMDFUL (G1STAT<1>) указывает, что буфер заполнен. Следующая команда, посланная в буфер, игнорируется и отбрасывается.
- Бит состояния уровня команд CMDLV (G1STAT<2>) показывает, что число команд в FIFO меньше запрограммированной величины в регистре уровня команд GCMDWMK<4:0> (G1CON1<12:8>).

Состояние буфера очереди команд может программироваться, указанием определенного числа команд, при достижении которого состояние буфера изменяется. Число команд устанавливается в регистре GCMDWMK<4:0>. Когда число команд в буфере FIFO становится меньше чем установленный уровень, устанавливается флаг прерывания уровня команд CMDLVIF (G1IR<2>) при условии, что разрешено прерывание CMDLVIE (G1IE<2>).

На [рисунок 16](#) показано восемь случаев изменения состояния уровня команд в буфере FIFO. В этом примере установленный уровень команд GCMDWMK<4:0> = 4. При загрузке команды в буфер FIFO, в зависимости от числа команд в буфере FIFO, устанавливаются или сбрасываются различные флаги состояния:

1. Буфер пустой. Прерывания CMDLVIF, CMDFULIF и CMDMPTIF запрещены.
2. Число команд (GCMDCNT<4:0>) увеличилось до 3. В этой точке прерывания всегда разрешены. Поскольку биты GCMDWMK = 4, состояние CMDLV остается равным '1'. Прерывание CMDLVIF не переключается, так как число команд не увеличилось до установленного уровня команд.
3. Когда число команд станет равным 4, состояние CMDLV сбрасывается в 0. Если число команд становится равным 16, устанавливается флаг прерывания заполнения буфера CMDFULIF.
4. Модули GPU начинают использовать команды, когда в буфере появляется первая команда. Если команды не добавляются, счетчик уменьшается после выполнения команды. В этой точке уровень команд равен установленному уровню, и флаг CMDLV остается сброшенным.
5. Счетчик команд уменьшился до 3. Прерывание CMDLVIF переключилось, так как счетчик стал меньше установленного уровня команд.
6. Модули GPU продолжают выполнять команды, и счетчик GCMDCNT становится равным 0, бит CMDMPTIF переключается, бит CMDLV сохраняет свое значение. Бит CMDLVIF не переключается, поскольку число команд остается меньше установленного уровня.

7. Число команд опять достигло 4. Флаг CMDLV сбрасывается, так как уровень команд достиг установленного уровня.
8. Модуль GPU выполнил команду, и счетчик стал равным 3. Флаг CMDLVIF переключился, так как число команд стало меньше установленного уровня.

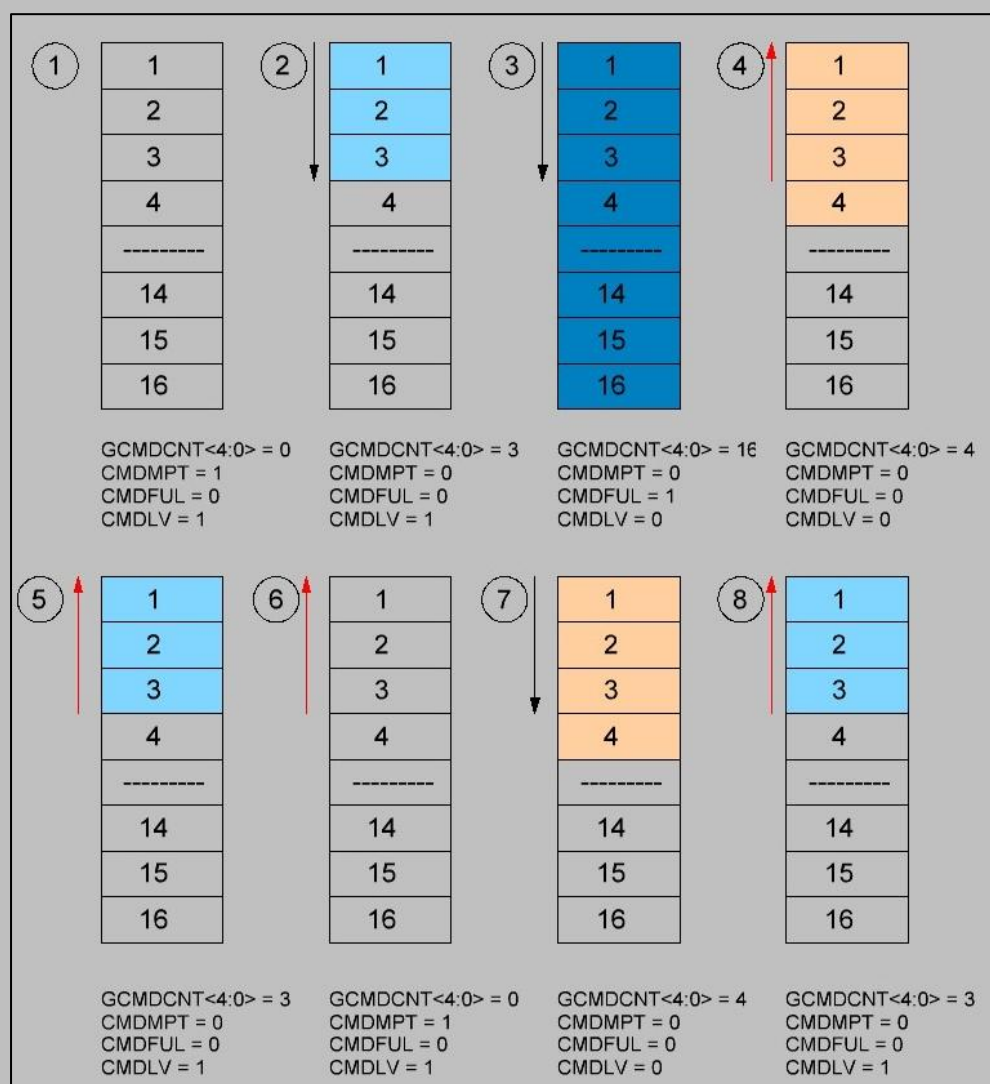


рисунок 16

На [рисунок 17](#) показана временная диаграмма генерации прерывания в зависимости от уровня команд в буфере.

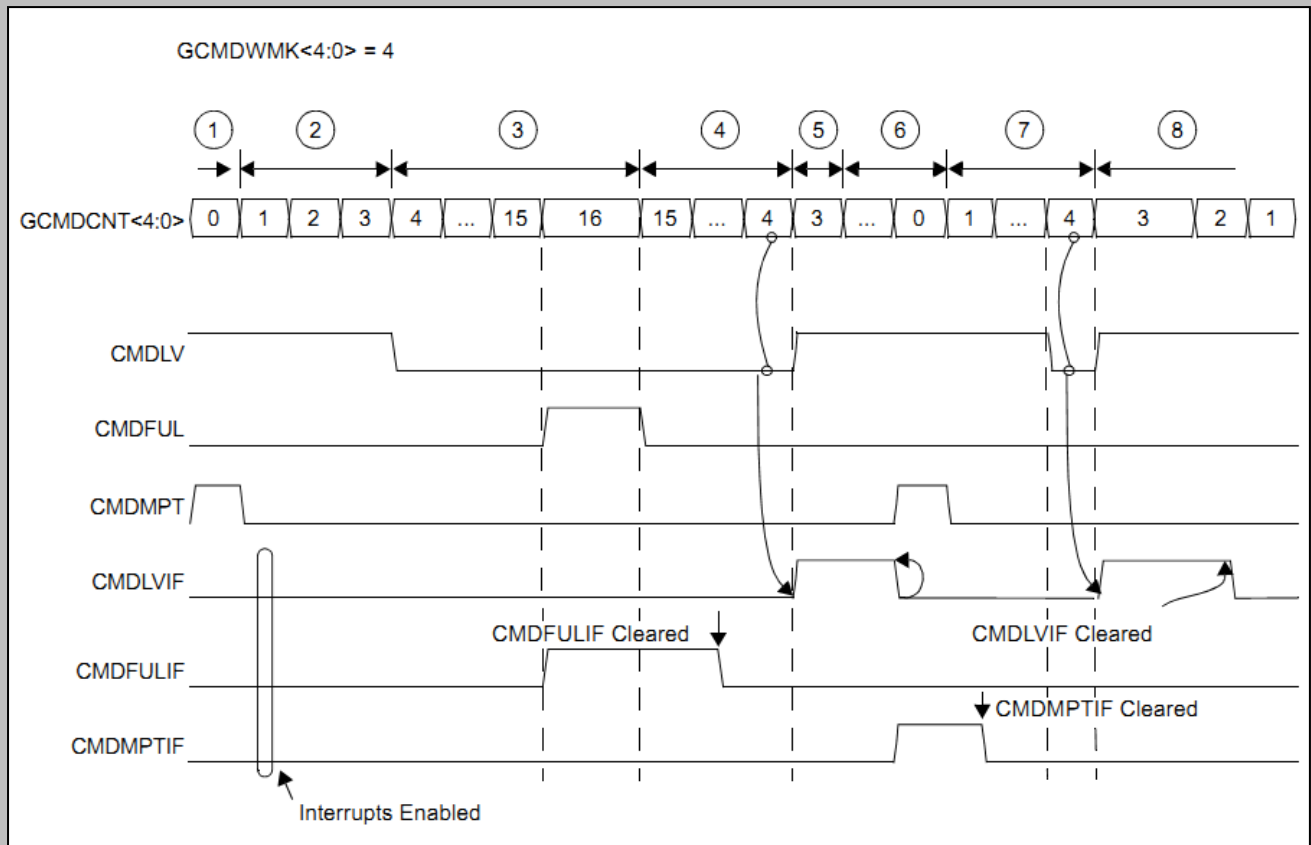


рисунок 17

Модуль обработки графических символов (CHRGPU)

Модуль CHRGPU предназначен для ускорения отображения символов. Модуль хранит таблицу шрифтов в памяти и передает коды символов, используя команды GPU. Модуль использует рабочую область 1 со стартовым адресом, указанным в W1ADR. Стартовый адрес определяется в регистрах G1W1ADRL и G1W1ADRH. Ширина и высота рабочей области 1 указывается в битах PUW (G1PUW<10:0>) и PUN (G1PUN<10:0>), соответственно.

В памяти может быть определено несколько таблиц шрифтов. Выбор таблицы производится по команде CHR_FONTBASE. Каждая таблица должна иметь символы одинаковой высоты, ширина символов может быть разной. Разные таблицы могут иметь символы разной высоты. Все символы в рабочей области 1 используют цвет фона и цвет переднего плана, также может использоваться прозрачность. Каждый бит символа представляется одним пикселем. Это означает, что при значении бита '1' будет использоваться цвет переднего плана, а при значении '0' – цвет фона. Если бит прозрачности установлен в команде CHR_PRINTCHAR, тогда цвет фона не будет отображен в памяти или рабочей области.

В рабочей области 1 площадь текста, на которой разрешено отображение символов, указывается в кодах команд CHR_TEXTAREASTART и CHR_TEXTAREAEND. Внутри текстовой площади левый верхний угол символа указывается в коде команды CHR_SETPRINTPOS. Любая часть символа, выпадающая из текстовой площади, будет отсекается.

PIC24FJ256DA210. Графический модуль

Формат таблицы шрифтов

Таблица шрифта состоит из следующих частей:

1. Заголовок
2. таблица, содержащая смещение битовой карты каждого символа и его ширину
3. Битовая карта всех символов

таблица 10 показывает структуру таблицы шрифта, в таблица 12 показаны поля заголовка таблицы шрифтов.

таблица 11

Смещение в байтах(offset)	Имя	15 14	13 12	11 10 9 8	7 6 5 4 3 2 1 0
Заголовок таблицы шрифта					
0x00		Reserved	Orient	Reserved	Font ID
0x02		First Character ID			
0x04		Last Character ID			
0x06		Reserved			Height
таблица символов					
0x08	1 st Character	Offset <7:0>			Width
0x0A		Offset <23:16>			Offset <15:8>
0x0C	2 nd Character	Offset <7:0>			Width
0x0E		Offset <23:16>			Offset <15:8>
.....					
offset _{last} - 4	Last Character	Offset <7:0>			Width
offset _{last} = (ID _{last} – ID ₁) + 1		Offset <23:8>			Offset <15:8>
Битовая карта символов					
1 character offset	1 st Character	1 st Character Bit Map Data			
.....					
2 character offset	2 nd Character	2 nd Character Bit Map Data			
.....					
.....					
Last character offset	Last Character	Last Character Bit Map Data			
.....					

таблица 12

Поле	Биты	Описание
Orient	2	Ориентация символа: 0 = нормальная 1 = поворот на 270 градусов 10 = поворот на 180 градусов 11 = поворот на 90 градусов
Font ID	8	Определяется пользователем. CHRGPU игнорирует это значение
First Character ID	16	ID первого символа.
Last Character ID	16	ID последнего символа.
Height	8	Высота всех символов в пикселях.
Width	8	Ширина символа, исключая биты добавленные для выравнивания на границу байта.
Offset	24	Смещение в байтах от начала заголовка до начала битовой карты символов.

таблица символов определяет ширину каждого символа и его расположение, которое задается смещением в байтах от начала заголовка шрифта. таблица символов должна содержать все коды символов от ID первого символа до ID последнего символа. В некоторых случаях используются не все символы, определенные в диапазоне ID. Для экономии памяти эти символы могут не иметь битовой карты, но поскольку все входы таблицы символов должны быть правильными, то для неиспользуемых символов ширина устанавливается в 0 и битовая карта такого символа обычно отображает символ ошибки (небольшой прямоугольник). Можно вообще удалить неиспользуемые символы из таблицы, но тогда следует предусмотреть программную перекодировку или блокировку вызова неиспользуемых символов. Этот способ также может сэкономить память.

Битовая карта символов может начинаться с адреса любого байта и каждая строка битовой карты должна начинаться с адреса байта. В зависимости от ориентации шрифта битовая карта символов может храниться в памяти различно. В поле шрифта указывается ориентация шрифта. Ориентация шрифта используется, когда дисплей монтируется с поворотом или требуется поворот текста. Ширина и высота символов при повороте не изменяется. Как уже говорилось, при повороте дисплея необходим поворот шрифта, что влечет за собой преобразование битовой карты символа при записи в дисплейный буфер. Учитывая это, каждая строка битовой карты символа будет иметь два размера: один для поворота на 0° и 180° и второй для поворота на 90° and 270°. Размер строки используется для вычислений при сохранении битовой карты в памяти. Поскольку каждая линия битовой карты символа должна начинаться на границе байта, необходимо дополнять битовую карту символа, когда размер строки не кратен 8. При повороте на 90° и 270° ширина символа не эквивалентна размеру строки. На рисунках показаны преобразования символов при поворотах. рисунок 18 показывает пример битовой карты для угла 0° и высоты символов в 7 пикселей. Три символа имеют ширину 11, 7 и 14 пикселей. Первый символ дополняется 5 битами для каждой строки. рисунок 19 показывает пример для угла поворота на 90°. Размер строки для всех трех символов одинаков. Каждая строка символа дополняется одним битом до границы байта.

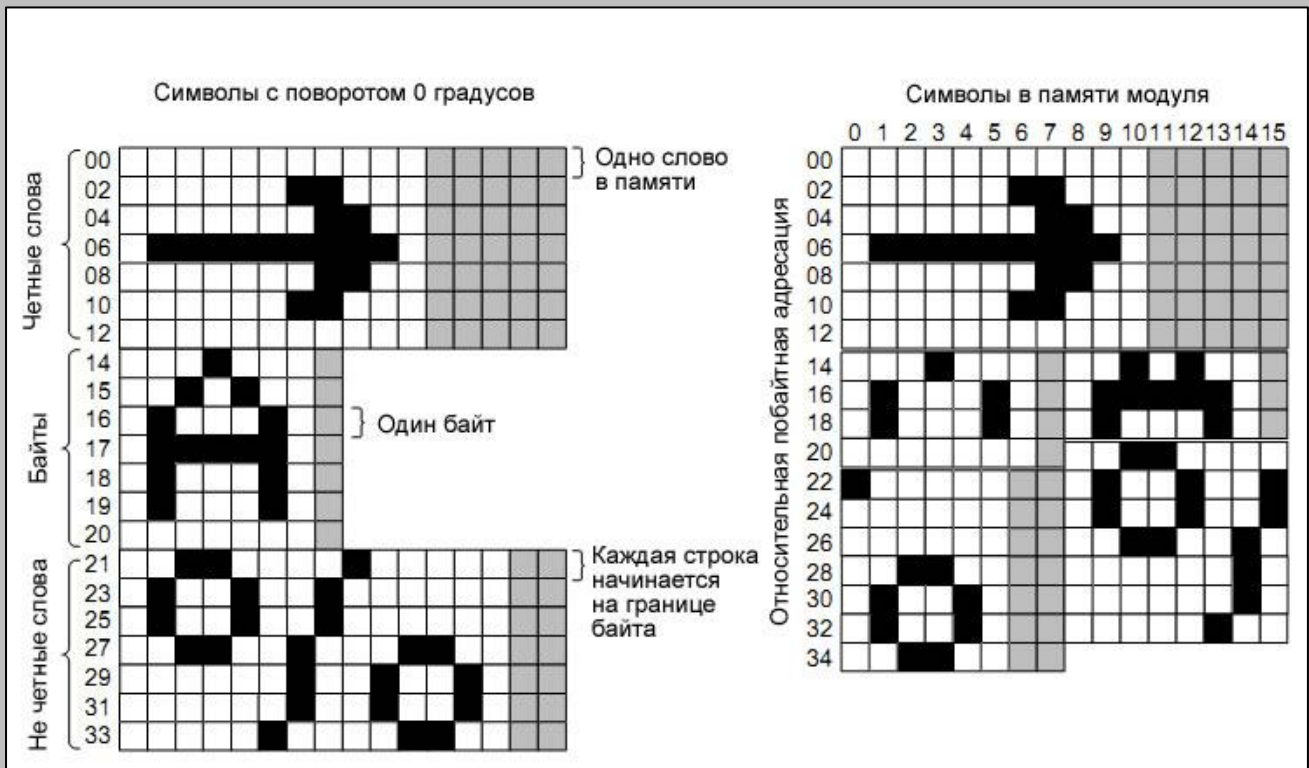


рисунок 18



рисунок 19

Команды модуля CHRGPU

Управление шрифтами производится с помощью семи команд.

CHR_FGCOLOR – установка цвета переднего плана

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:16>	GCMD<15:0> Color Value
0x5	0x0	Reserved	Color / index CLUT

CHR_BGCOLOR – установка цвета фона

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:16>	GCMD<15:0> Color Value
0x5	0x1	Reserved	Color / index CLUT

CHR_FONTBASE – установка базового адреса таблицы шрифтов

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:0> Address
0x5	0x2	Адрес байта, выровненный на границу слова

CHR_PRINTCHAR – отображает символ, используя код символа для входа в битовую карту символа. При переходе на новую строку производится регулировка отображения в соответствии с высотой символов, указанной в таблице шрифта. При установке бита Tr цвет фона не рисуется. Если отображение символа выходит за границы текстового прямоугольника, то отображается только часть символа, включая границу прямоугольника.

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23> Tr	GCMD<22:16>	GCMD<15:0> Character Code
0x5	0x3	Бит прозрачности	Reserved	Код отображаемого символа

PIC24FJ256DA210. Графический модуль

CHR_TXTAREASTART — устанавливает координаты левого верхнего угла текстового прямоугольника, где разрешено отображение.

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:12> X coordinate	GCMD<11:0> Y coordinate
0x5	0x8	Координата X верхнего левого угла текстовой площади	Координата Y верхнего левого угла текстовой площади

CHR_TXTAREAEND — устанавливает координаты нижнего правого угла текстового прямоугольника, где разрешено отображение.

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:12> X coordinate	GCMD<11:0> Y coordinate
0x5	0x9	Координата X нижнего правого угла текстовой площади	Координата Y нижнего правого угла текстовой площади

CHR_PRINTPOS — устанавливает позицию следующего отображаемого

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:12> X coordinate	GCMD<11:0> Y coordinate
0x5	0xA	Координата X верхнего левого угла следующего символа	Координата Y верхнего левого угла следующего символа

Операции модуля CHRGPU

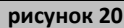
Для отображения текста следует выполнить следующие шаги:

1. Сохранить одну или несколько таблиц шрифтов в памяти.
2. Установить адрес таблицы шрифта, используя команду CHR_FONTBASE.
3. Установить размеры площади текста, где он будет отображаться, определяемые командами CHR_TXTAREASTART и CHR_TXTAREAEND.
4. Установить цвет переднего плана и фона, используя команды CHR_FGCOLOR и CHR_BGCOLOR.
5. Установить позицию вывода, используя команду CHR_PRINTPOS.
6. Напечатать текст, используя команду CHR_PRINTCHAR.

Модуль CHRGPU работает только с рабочей областью 1, поэтому логично сделать ширину области 1 равной ширине дисплейной области, чтобы пиксели текста всегда соответствовали пикселям дисплейной области.

рисунок 20 демонстрирует пример отображения дисплейного буфера, объявленного как рабочая область 1. Можно объявить рабочую область 1 вне дисплейного буфера, но при этом очень важно сделать ее ширину равной ширине дисплейного буфера. Для отображения отдельной рабочей области ее следует скопировать в дисплейный буфер. Если ширина рабочей области не равна ширине дисплейного буфера, код приложения должен учитывать разницу при копировании, иначе изображение будет искажено.

Отображение длинной строки по координате X или Y будет выполнено даже при превышении размеров дисплея. Когда счетчик координаты достигает конца координаты, он начинает счет с нуля и вывод строки продолжится в соответствии с установками для вывода текста. Программа может проверить текущую позицию печати в регистрах G1CHRX и G1CHRY, для определения выхода за пределы размера дисплея.



Существуют три типа операций копирования:

- На **рисунок 21** показаны все три операции копирования.

В зависимости от типа операции данные источника и адресата копируются, модифицируются или игнорируются.

Стр. 36

PIC24FJ256DA210. Графический модуль

данных. Модуль RCCGPU всегда выполняет операции с каждым пикселем индивидуально, несмотря на то, что память адресуется пословно и в слове данных может быть несколько пикселей.

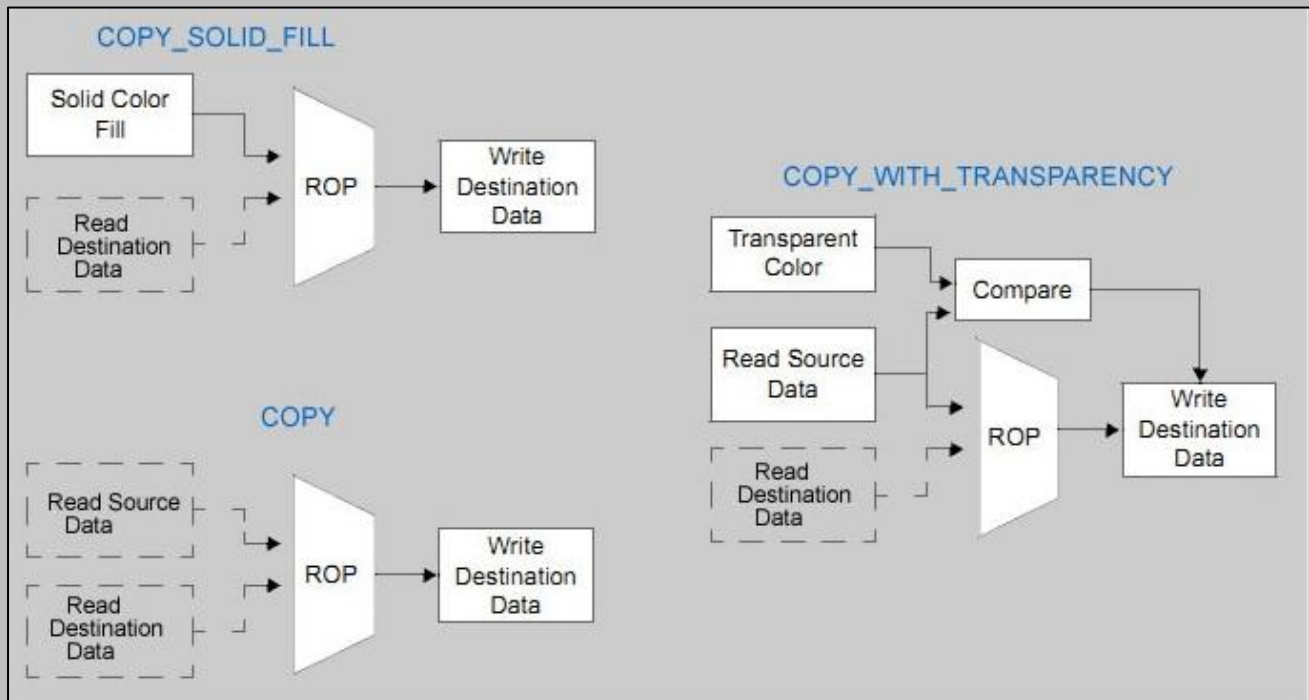


рисунок 21

В [таблица 13](#) перечислены команды ROP и действия, выполняемые с источником и получателем данных.

таблица 13

Name	Value	Source (S) and Destination (D) Logical Operation
ROP_0	0000	0 (Black)
ROP_1	0001	not (S or D)
ROP_2	0010	(not S) and D
ROP_3	0011	not S
ROP_4	0100	S and (not D)
ROP_5	0101	not D
ROP_6	0110	S XOR D
ROP_7	0111	not (S and D)
ROP_8	1000	S and D
ROP_9	1001	not (S XOR D)
ROP_A	1010	D
ROP_B	1011	(not S) or D
ROP_C	1100	S
ROP_D	1101	S or (not D)
ROP_E	1110	S or D
ROP_F	1111	1 (White)

Операции COPY и COPY_WITH_TRANSPARENCY всегда разрешены для источника данных. Операция ROP_0 или ROP_F не применяется для пикселей источника с цветом таким же, как прозрачный цвет. Это поведение может использоваться для стирания части рабочей области с черным или белым цветом, не затрагивая области с прозрачным цветом. Эффект использования операций ROP может быть различен в зависимости от того используется или нет цветовая таблица (CLUT). Если CLUT запрещена, и например, выбрана операция ROP_E, тогда логическая операция OR над красным и голубым цветом дает пурпурный (magenta) цвет. Если CLUT разрешена тогда операция ROP_E над пикселем 4 и пикселем 2 даст результат 6, это будет цвет в CLUT по адресу 6.

PIC24FJ256DA210. Графический модуль

Команды модуля RCCGPU

RCC_SRCADDR— устанавливает смещения адреса исходного прямоугольника.

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:0>
0x6	0x2	Смещение адреса исходного прямоугольника

RCC_DESTADDR— устанавливает смещения адреса конечного прямоугольника.

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:0>
0x6	0x3	Смещение адреса конечного прямоугольника

RCC_RECTSIZE— устанавливает высоту и ширину копируемого прямоугольника.

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23>	GCMD<22:12>	GCMD<11>	GCMD<10:0> Character Code
0x6	0x4	Reserved	Ширина	Reserved	Высота

RCC_COLOR— установка цвета для использования в операции ROP, таких как сплошная заливка и прозрачность

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:16>	GCMD<15:0> Color Value
0x6	0x6	Reserved	Цвет

RCC_STARTCOPY— выполняет копирование с операцией ROP.

• ST – Тип адреса источника:

0 = адресация источника дискретна. Эта установка используется, когда прямоугольник находится в памяти дисплейного буфера (отображается на экране). В этом случае адресация памяти разрывается на участке между самым правым пикселем и самым левым на следующей строке.

1 = Адресация источника непрерывна. Эта установка используется, когда прямоугольник находится в памяти вне дисплейного буфера. Например, блокнотная память оперирует с данными перед отображением на дисплее. В этом случае прямоугольник храниться в памяти в более компактном виде, нет разрывов адресации между самым правым пикселем и самым левым на следующей строке.

• DT – Тип адреса получателя. Аналогично описанию для ST.

• ROP – Выполняемая растровая операция.

• OPER – операция, выполняемая модулем RCCGPU.

000 = Сплошная заливка с ROP

001 = Копирование с ROP

110 = Копирование с ROP и прозрачность.

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:10>	GCMD<9:7>	GCMD<6:3>	GCMD<2>	GCMD<1>	GCMD<0>
0x6	0x7	Reserved	Операция	ROP	DT	ST	Reserved

Примеры

Следующие примеры показывают, как модуль RCCGPU генерирует простые объекты. Рабочая область 1 (W1ADR<23:0>) всегда устанавливается как источник, а рабочая область 2 как получатель.

Точка в один пиксель

- Установить цвет пикселя (RCC_COLOR)
- Установить смещение адреса получателя (RCC_DESTADDR)
- Установить ширину и высоту равную 1 (RCCRECTSIZE)
- Выполнить команду RCC_STARTCOPY с OPER = 000, ROP = ROP_C, DT = тип адреса получателя.

Сплошная цветная вертикальная/горизонтальная линия в один пиксель

- Установить цвет линии (RCC_COLOR). Выбранный цвет будет служить как источник данных, рабочая область 1 игнорируется.
- Установить смещение адреса получателя. Это будет адрес верхнего пикселя для вертикальной линии или адрес левого пикселя для горизонтальной линии (RCC_DESTADDR)
- Для вертикальной линии: Установить ширину прямоугольника равную 1 и высоту равную длине линии (RCC_RECTSIZE)
- Для горизонтальной линии: Установить ширину равную длине линии и высоту равную 1 (RCC_RECTSIZE)
- Выполнить команду RCC_STARTCOPY с OPER = 000, ROP = ROP_C, DT = тип адреса получателя; Источник будет игнорироваться в этой операции, поскольку нет выбранной ROP и OPER.

Прямоугольник

Чтобы сделать прямоугольник с линиями шириной в n-пикселей следует просто создать четыре сплошные линии сторон прямоугольника.

Сплошные прямоугольники

- Установить цвет (RCC_COLOR). Выбранный цвет будет служить как источник данных, рабочая область 1 игнорируется.
- Установить смещение адреса получателя. Это будет адрес пикселя, который определяет верхний левый угол прямоугольника (RCC_DESTADDR).
- Установить ширину и высоту прямоугольника (RCC_RECTSIZE).
- Выполнить RCC_STARTCOPY с OPER = 000, ROP = ROP_C, DT = тип адреса получателя.

Копирование прямоугольной области в другое расположение

- Установить смещение адреса источника, это будет адрес пикселя, который определяет верхний левый угол источника (RCC_SRCADDR).
- Установить смещение адреса получателя это будет адрес пикселя, который определяет верхний левый угол потребителя (RCC_DESTADDR).
- Выполнить RCC_STARTCOPY с OPER = 001, ROP = ROP_C, ST и DT = типы адресов источника и получателя.

Копирование диапазона непрерывных данных

- Установить смещение адреса первого копируемого байта источника (RCC_SRCADDR).
- Установить смещение адреса получателя первого байта (RCC_DESTADDR).
- Установите размеры прямоугольника высота равную 1 и ширину равную числу копируемых байт * 8 / бит на пиксель (RCC_RECTSIZE), где результат деления округляется до следующего верхнего адреса.
- Выполнить RCC_STARTCOPY с OPER = 001 и ROP = ROP_C; ST и DT не имеет здесь значения, поскольку операция выполняется с высотой равной 1.

Модуль декомпрессии данных (IPU)

Модуль IPU предназначен для декомпрессии данных сжатых по алгоритму Хаффмана с постоянными кодами. Динамические коды Хаффмана не поддерживаются. Модуль IPU использует две рабочие области, первая для сжатых данных, вторая для декомпрессированных данных. Обе рабочие области инициализируются с помощью регистров W1ADR и W2ADR. Эта функция особенно полезна в приложениях с ограниченной памятью. Например, можно иметь несколько разных сжатых таблиц шрифтов и по мере необходимости распаковывать их в одну область памяти для использования.

Для декомпрессии данных следует выполнить следующие шаги:

1. Инициализировать рабочие области 1 и 2 с помощью регистров W1ADR и W2ADR.
2. Установить смещение адреса для сжатых данных в рабочей области 1, используя команду IPU_DESTADDR. Эффективный адрес ($\text{Address} = \text{Work Area 2} + \text{DSTOFFSET}$) должен указывать на начало сжатых данных.
3. Установить смещение адреса для декомпрессированных данных, используя команду IPU_DESTADDR. Эффективный адрес получателя $\text{Address} = \text{Work Area 1} + \text{SRCOFFSET}$.
4. Выполнить команду декомпрессии IPU_DECOMPRESS, указав размер не сжатых данных (DSTSIZE) в байтах.
5. Определить окончание операции, используя регистр состояния G1IPU.

Команды IPU

IPU_SRCADDR – устанавливает смещения адреса первого сжатого байта от начала 1-й рабочей области.

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:0>
0x7	0x1	Смещение адреса

IPU_DESTADDR – устанавливает смещения адреса первого не сжатого байта от начала 2-й рабочей области.

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:0>
0x7	0x2	Смещение адреса

IPU_DECOMPRESS – выполнение декомпрессии с указанием размера декомпрессированных данных.

GCMD<31:28> CHRGPU Address	GCMD<27:24> Command	GCMD<23:0>
0x7	0x4	Размер в байтах

Хотя команды IPU разрешают декомпрессировать данные, начиная на границе байта, все же декомпрессированные данные лучше располагать на границе слова. Например, это может быть таблица шрифтов или изображение декомпрессированное прямо в дисплейный буфер.

Прерывания

Графический контроллер формирует девять прерываний, четыре для операций GPU, два для гасящих импульсов и три для буфера команд FIFO. Каждый источник прерывания имеет бит состояния и бит разрешения. Все девять прерываний объединяются в один сигнал прерывания от графического контроллера (рисунок 22). Подпрограмма обработки прерывания от графического контроллера должна определять источник события и обслуживать его. Разрешение прерывания устанавливаются в регистре G1IE, состояние проверяется в регистре G1IR. Если прерывание не разрешено, состояние события может быть также проверено в регистре G1STAT.

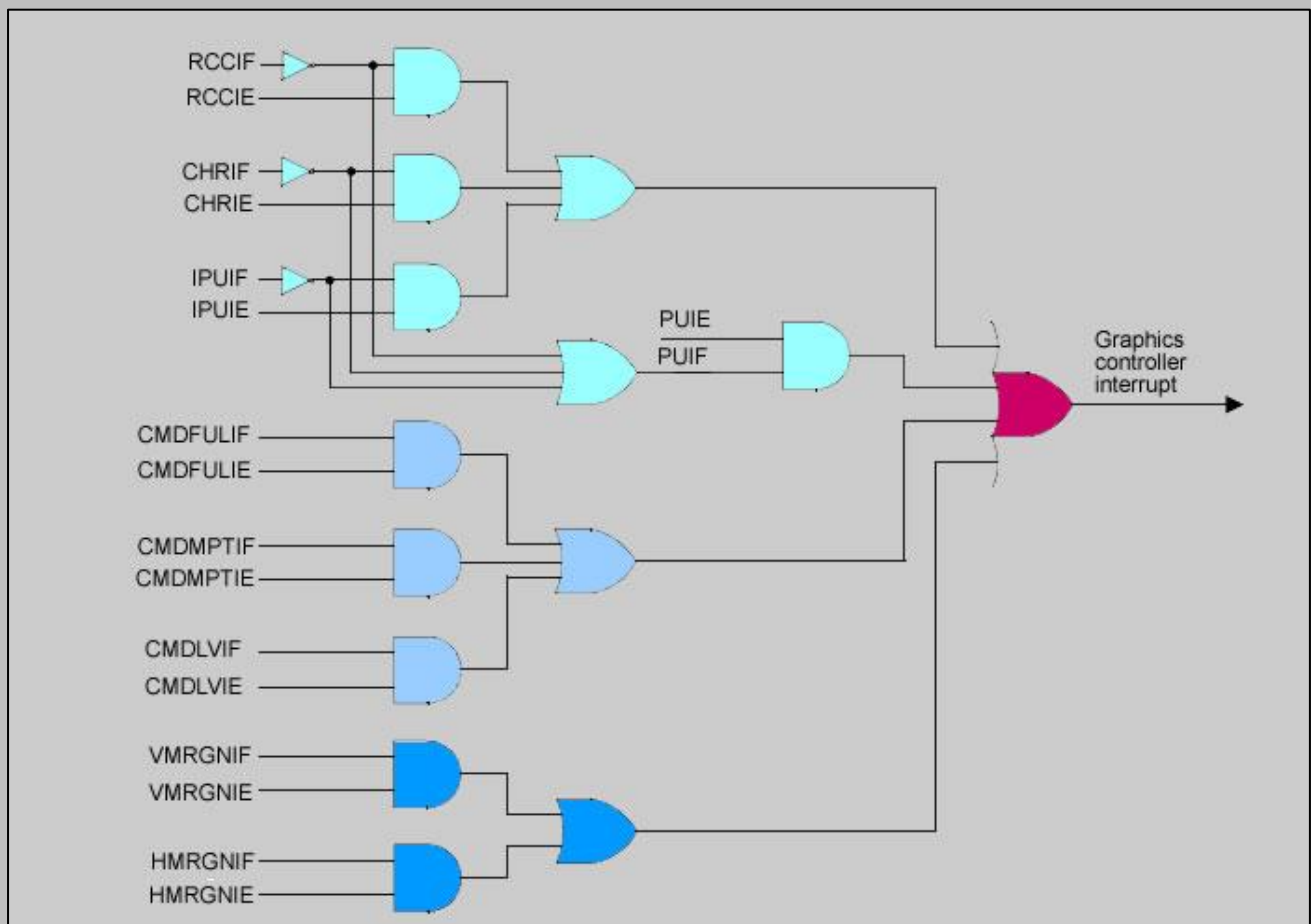


рисунок 22

Прерывания GPU

Поскольку только один модуль GPU может выполнять команду в любой момент времени, каждому GPU назначено свое прерывание. Приложение может использовать IPUIF, RCCIF и CHRIF для определения простоя каждого GPU и сообщить приложению о готовности выполнять новую команду. Общая занятость шины GPU доступна через PUIF. Это прерывание вызывается, когда один из GPU выполняет команду, в отличие от остальных прерываний, которые вызываются, в состоянии простоя GPU.

Прерывания буфера команд FIFO

Прерывания буфера команд FIFO описаны в разделе "Состояние очереди команд FIFO".

Прерывания синхроимпульсов

В режиме TFT, в период действия синхронизирующих импульсов, обновление экрана не производится и можно обращаться к дисплейному буферу или, например, модифицировать цветовую таблицу CLUT. Прерывание для вертикального синхроимпульса VMRGNIF разрешается битом VMRGNIE (G1IE<4>), прерывание для горизонтального синхроимпульса PMRGNIF разрешается битом PMRGNIE (G1IE<3>).

Последовательность подачи питания дисплея

Дисплейный контроллер обеспечивает определенную последовательность подачи сигнала на включение/выключение питания дисплея и синхроимпульсов. Сигналы, формируемые дисплейным контроллером, показаны на [рисунок 23](#). Для дисплеев, которые имеют несколько напряжений питания, нужно организовывать внешние цепи подачи дополнительного питания.

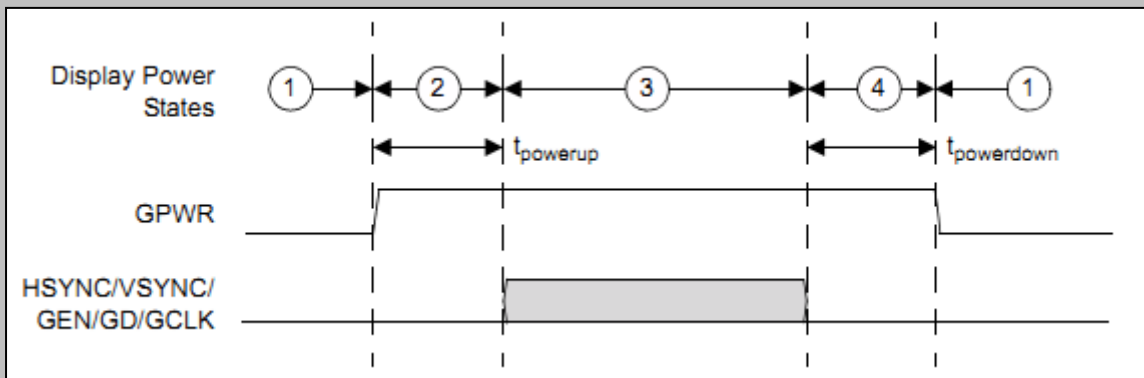


рисунок 23

Формирование последовательности подачи питания разрешается при установке бита DPPOWER (G1CON3<8>). Сигнал GPWR генерируется после некоторой задержки, вызванной привязкой к тактовой частоте GLCK. Бит DPPOWER может быть сброшен в любое время, питание дисплея снимается, все сигналы становятся не активными. Время включения и выключения (состояние 2 и 4 на рисунке 23) может быть определено по формуле.

$t_{powerup}$ или $t_{powerdown} = G1CLUTWR<15:0> * 64 \text{ GCLK Cycles}$

Перед установкой или сбросом бита DPPOWER в регистр G1CLUTWR должна быть записана правильная величина, которая определена в спецификации дисплея. Использование сигналов GPWR и GEN не является обязательным, можно использовать и внешние сигналы.

Операции в режиме энергосбережения

Семейство PIC24F имеет три режима питания, нормальный режим и два режима энергосбережения. Режимы энергосбережения вызываются по команде процессора PWRSV. Операции модуля зависят от выбранного режима энергосбережения.

Спящий режим

В спящем режиме все тактовые частоты графического модуля запрещены. Никакие операции не выполняются. Дисплей перед переводом в спящий режим должен быть выключен, иначе возможно его повреждение.

Ждущий режим

Переводом в ждущий режим управляет бит G1SIDL(G1CON1 <13>). При установке G1SIDL = 1 поведение модуля подобно спящему режиму.

Воздействие сигнала сброса

При сбросе все регистры графического модуля устанавливаются в состояние по умолчанию.

Очередь команд FIFO очищается, и любая исполняемая команда прерывается с возвратом в состояние простоя. Если питанием дисплея управляет модуль, то с помощью сигнала GPWR можно определить состояние питания дисплея. При выполнении сброса приложение должно правильно выполнить последовательность включения и выключения дисплея.

Модуль графического контроллера по сигналу сброса не устанавливает состояние рабочих областей и состояние дисплейного буфера.

Сброс устройства (MCLR)

Сброс устройства приводит к сбросу всех регистров графического контроллера. Модуль выключается.

Сброс при включении питания (POR)

Сброс устройства приводит к сбросу всех регистров графического контроллера. Модуль выключается.

Сброс по сторожевому таймеру (WDT)

Сброс по сторожевому таймеру приводит к сбросу всех регистров графического контроллера. Модуль выключается.

Регистры модуля

Модуль GFX включает большое количество управляющих регистров и регистров состояния. Однако программисту, использующему графическую библиотеку, практически нет нужды программировать регистры модуля. Только при внедрении в разработку нового драйвера дисплея придется детально изучить программирование модуля, но и в этом случае можно ограничиться регистрами, формирующими развертку дисплея. Остальные части нового драйвера могут быть перенесены из готовых драйверов библиотеки.

Общие управляющие регистры

Регистр команд GPU

G1CMDL и **G1CMDH** – младший и старший регистр команд модуля (GPU). Эти регистры формируют 32-разрядное командное поле. Команды записываются в эти регистры из буфера команд FIFO. Полная 32-битная команда определяется регистрами G1CMDH и G1CMDL. Слово, записанное в регистр G1CMDH, загружает новую команду буфер FIFO. Запись байта в регистр G1CMDH разрешается, но только запись старшего байта загружает новую команду буфер FIFO. Запись младшего байта только обновляет разряды регистра.

Примечание

Здесь и далее по тексту используются следующие сокращения:

- R – читаемый бит.
- W – записываемый бит.
- U – не реализованный бит, читается как 0.
- -n – величина устанавливаемая после сброса микропроцессора.
- 1 – бит установлен, 0 – бит сброшен, x – бит неизвестен.

Регистр G1CMDL

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCMD15	GCMD14	GCMD13	GCMD12	GCMD11	GCMD10	GCMD9	GCMD8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCMD7	GCMD6	GCMD5	GCMD4	GCMD3	GCMD2	GCMD1	GCMD0
bit 7							bit 0

Регистр G1CMDH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCMD31	GCMD30	GCMD29	GCMD28	GCMD27	GCMD26	GCMD25	GCMD24
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCMD23	GCMD22	GCMD21	GCMD20	GCMD19	GCMD18	GCMD17	GCMD16
bit 7							bit 0

PIC24FJ256DA210. Графический модуль

Управляющие регистры дисплея G1CON1 - G1CON3

- **G1CON1** – Первый управляющий регистр дисплея. Этот регистр разрешает работу модуля GFX, устанавливает уровень прерывания для определенного количества команд в очереди команд FIFO, при достижении которого будет происходить прерывание, устанавливает цветовую глубину. Регистр также содержит состояние текущей очереди команд FIFO (16 команд).
- **G1CON2** – Второй управляющий регистр дисплея. Этот регистр устанавливает тип дисплея, подключаемого к модулю, указывает цветовую глубину, в соответствии с установками G1CON1, управляет отображаемыми данными для уменьшения шума переключения и отображает тестовый шаблон в режиме отладки.
- **G1CON3** – Третий управляющий регистр дисплея. Регистр управляет питанием дисплея и параметрами синхроимпульсов. Разрешает выдачу сигналов синхронизации и данных.

Регистр G1CON1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G1EN	—	G1SIDL	GCMDWMK4	GCMDWMK3	GCMDWMK2	GCMDWMK1	GCMDWMK0
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0
PUBPP2	PUBPP1	PUBPP0	GCMDCNT4	GCMDCNT3	GCMDCNT2	GCMDCNT1	GCMDCNT0
bit 7							bit0

bit 15 G1EN: Бит разрешения модуля

1 = Отображение разрешено

0 = Отображение запрещено

bit 13 G1SIDL: Стор в режиме простоя (Idle)

1 = Отображение останавливается в режиме простоя

0 = Отображение не останавливается в режиме простоя

bit 12-8 GCMDWMK<4:0>: Биты вызова прерывания по числу команд (Watermark) в буфере FIFO.

Устанавливают число команд в буфере FIFO для вызова прерывания CMDLVIF и установки флага CMDLV.

GCMDWMK<4:0> > 10000 = Резервировано:

10000 = Если число команд в FIFO от 16 до 15 команд, the CMDLVIF будет вызываться прерывание и устанавливаться флаг CMDLV.

01111 = Если число команд в FIFO от 15 до 14 команд, the CMDLVIF будет вызываться прерывание и устанавливаться флаг CMDLV.

-
-
-

00001 = Если число команд в FIFO от 1 до 0 команд, the CMDLVIF будет вызываться прерывание и устанавливаться флаг CMDLV.

00000 = Прерывание не вызывается, флаг CMDLV не устанавливается

bit 7-5 PUBPP<2:0>: Число бит на пиксель GPU

100 = 16 бит на пиксель

011 = 8 бит на пиксель

010 = 4 бит на пиксель

001 = 2 бит на пиксель

000 = 1 бит на пиксель

PIC24FJ256DA210. Графический модуль

Остальные значения зарезервированы

bit 4-0 GCMDCNT<4:0>: Определяют число команд в буфере FIFO.

Когда буфер FIFO полный любая записываемая команда игнорируется.

10000 = 16 в буфере FIFO

01111 = 15 в буфере FIFO

-
-
-

00001 = 1 в буфере FIFO

00000 = 0 в буфере FIFO

Регистр **G1CON2**

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
DPGWDTH1	DPGWDTH0	DPSTGER1	DPSTGER0	—	—	DPTEST1	DPTEST0
bit 15							bit 8
R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
DPBPP2	DPBPP1	DPBPP0	—	—	DPMODE2	DPMODE1	DPMODE0
bit 7							bit 0

bit 15-14 DPGWDTH<1:0>: определяет разрядность данных дисплея STN.

11 = Резервировано

10 = 16 бит

01 = 8 бит

00 = 4 бит

Эти биты не имеют значения для TFT дисплея, шина данных которого всегда 16 бит.

bit 13-12 DPSTGER<1:0>: Биты синхронизации данных для уменьшения шума при одновременном переключении данных. Подробно в параграфе 43.4.4

11 = Задержка отображения данных в группе:

Bit Group 0: 0 4 8 12 – нет задержки

Bit Group 1: 1 5 9 13 – задержка на ½ цикла GPUCLK

Bit Group 2: 2 6 10 14 – задержка на полный цикл GPUCLK

Bit Group 3: 3 7 11 15 – задержка на 1½ цикла GPUCLK

10 = Четные биты задерживаются на 1 полный цикл GPUCLK, не четные биты не задерживаются.

01 = Не четные биты задерживаются на ½ цикла GPUCLK, четные биты не задерживаются.

00 = Все данные синхронизируются на одном фронте цикла GPUCLK.

bit 9-8 DPTTEST<1:0>: Биты тестового генератора шаблона.

Тестовый шаблон может использоваться для тестирования интерфейса дисплея без загрузки памяти интерфейса и буфера отображения.

11 = Границы

10 = Прямоугольник.

01 = Черный экран.

00 = Нормальный режим, тест отключен.

bit 7-5 DPBPP<2:0>: Число бит на пиксель

Эти установки должны соответствовать установкам в PUBPP<2:0> (G1CON1<7:5>).

100 = 16 бит на пиксель

011 = 8 бит на пиксель

010 = 4 бит на пиксель

PIC24FJ256DA210. Графический модуль

001 = 2 бит на пиксель

000 = 1 бит на пиксель

Остальные значения зарезервированы

bit 2-0 DPMODE<2:0>: Тип дисплея.

011 = Цветной STN

010 = Монохромный STN

001 = TFT

000 = Нет дисплея

Остальные значения зарезервированы.

Регистр **G1CON3**

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	DPPINOE	DPPOWER
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPCLKPOL	DPENPOL	DPVSPOL	DPHSPOL	DPPWROE	DPENOE	DPVSOE	DPHSOE
bit 7							bit0

bit 9 DPPINOE: Бит разрешения выходного контакта клавиатуры.

Бит разрешения выхода клавиатуры DPPINOE, также должны быть установлены дополнительные разрешения для клавиатуры GDBEN<15:0>, DPENOE, DPPWROE, DPVSOE и DPHSOE.

1 = Выход клавиатуры разрешен

0 = Выход клавиатуры не разрешен

Контакты, используемые сигналами, также присвоены следующему включенному модулю. Для управления данными сигналами можно использовать регистр GDBEN<15:0>.

bit 8 DPPOWER: бит управления питанием дисплея.

1 = Устанавливает управляющий порт GPWR в 1

0 = Устанавливает управляющий порт GPWR в 0

bit 7 DPCLKPOL: бит полярности тактовой частоты GCLK. Меняет фазу GCLK на 180°.

1 = Дисплей фиксирует данные по фронту GCLK

0 = Дисплей фиксирует данные по спаду GCLK

bit 6 DPENPOL: бит полярности сигнала разрешения GEN.

Для режима TFT (DPMODE (G1CON2<2:0>) = 001):

1 = Активный высокий уровень (GEN)

0 = Активный низкий уровень (GEN)

Для режима STN (DPMODE (G1CON2<2:0>) = 010 or 011):

1 = GEN включается как сигнал сдвига по тактовой частоте (Shift Clock mode)

0 = GEN подключается ко входу MOD дисплея (Line/Frame Toggle mode)

bit 5 DPVSPOL: бит полярности сигнала вертикальной синхронизации (VSYNC)

1 = Активный высокий уровень (VSYNC)

0 = Активный низкий уровень (VSYNC)

bit 4 DPHSPOL: бит полярности горизонтальной синхронизации (HSYNC)

1 = Активный высокий уровень (HSYNC)

0 = Активный низкий уровень (HSYNC)

PIC24FJ256DA210. Графический модуль

bit 3 DPPWROE: бит разрешения порта управления последовательностью подачи питания (GPWR).

1 = GPWR порт разрешен (контакт управляется битом DPPPOWER (G1CON3<8>))

0 = GPWR порт запрещен (контакт может быть использован для ввода/вывода)

bit 2 DPENOE: бит разрешения включения порта GEN.

1 = GEN порт разрешен

0 = GEN порт запрещен

bit 1 DPVSOE: бит разрешения порта вертикальной синхронизации.

1 = VSYNC порт разрешен

0 = VSYNC порт запрещен

bit 0 DPHSOE: бит разрешения порта горизонтальной синхронизации.

1 = HSYNC порт разрешен

0 = HSYNC порт запрещен

Регистр состояния G1STAT

Регистр содержит состояние GPUs, указывает состояние обратного хода развертки (вертикальной и горизонтальной), уровень прерывания FIFO.

R-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
PUBUSY	—	—	—	—	—	—	—
bit 15							bit 8
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
IPUBUSY	RCCBUSY	CHRBUSY	VMRGN	HMRGN	CMDLV	CMDFUL	CMDMPT
bit 7							bit 0

bit 15 PUBUSY: бит состояния занятости.

Бит логически эквивалентен комбинации битов IPUBUSY, RCCBUSY и CHRBUSY.

1 = По крайней мере один блок обработки занят

0 = Ни один блок обработки не занят

bit 7 IPUBUSY: бит занятости блока распаковки изображения

1 = IPU занят

0 = IPU свободен

bit 6 RCCBUSY: бит занятости блока обработки прямоугольников

1 = RCCGPU занят

0 = RCCGPU свободен

bit 5 CHRBUSY: бит занятости блока обработки символов

1 = CHRGPU занят

0 = CHRGPU свободен

bit 4 VMRGN: бит состояния обратного хода развертки по вертикали

1 = интерфейс в состоянии обратного хода развертки по вертикали

0 = интерфейс в состоянии прямого хода вертикальной развертки

PIC24FJ256DA210. Графический модуль

bit 3 HMRGN: бит состояния обратного хода развертки по горизонтали

1 = интерфейс в состоянии обратного хода развертки по горизонтали

0 = интерфейс в состоянии прямого хода горизонтальной развертки

bit 2 CMDLV: бит состояния указателя уровня команд стека FIFO

1 = команд в стеке FIFO меньше чем величина, установленная в GCMDWMK

0 = число команд в стеке FIFO равно или больше установленной в GCMDWMK величины

bit 1 CMDFUL: бит заполнения стека команд FIFO

1 = стек команд FIFO заполнен

0 = стек команд FIFO пустой

bit 0 CMDMPT: бит опустошения стека команд FIFO

1 = стек команд FIFO пустой

0 = стек команд FIFO не пустой

Регистр разрешения прерываний G1IE

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
PUIE	—	—	—	—	—	—	—
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IPUIE	RCCIE	CHRIE	VMRGNIE	HMRGNIE	CMDIVIE	CMDFULIE	CMDMPTIE
bit 7							bit 0

bit 15 PUIE: бит разрешения работы блока прерываний

1 = прерывания разрешены

0 = прерывания запрещены

bit 7 IPUIE: бит разрешения выполнения прерывания от модуля распаковки

1 = выполнение прерываний IPU разрешено

0 = выполнение прерываний IPU запрещено

bit 6 RCCIE: бит разрешения выполнения прерывания от блока прямоугольников

1 = выполнение прерываний RCCGPU разрешено

0 = выполнение прерываний RCCGPU запрещено

bit 5 CHRIE: бит разрешения выполнения прерывания от блока обработки символов

1 = выполнение прерываний CHRGPU разрешено

0 = выполнение прерываний CHRGPU запрещено

bit 4 VMRGNIE: бит разрешения выполнения прерывания по сигналу обратного хода вертикальной развертки

1 = прерывание разрешено на обратном ходе развертки

0 = прерывание запрещено на обратном ходе развертки

bit 3 HMRGNIE: бит разрешения выполнения прерывания по сигналу обратного хода горизонтальной развертки

1 = прерывание разрешено на обратном ходе развертки

0 = прерывание запрещено на обратном ходе развертки

bit 2 CMDLVIE: бит разрешения выполнения прерывания по уровню команд в стеке команд FIFO

PIC24FJ256DA210. Графический модуль

1 = прерывание разрешено

0 = прерывание запрещено

bit 1 CMDFULIF: бит разрешения выполнения прерывания по заполнению стека команд FIFO

1 = прерывание разрешено при заполнении стека FIFO

0 = прерывание запрещено при заполнении стека FIFO

bit 0 CMDMPTIE: бит разрешения выполнения прерывания по опустошению стека команд FIFO

1 = прерывание разрешено при опустошении стека FIFO

0 = прерывание запрещено при опустошении стека FIFO

Регистр состояния прерываний G1IR

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
PUIF	—	—	—	—	—	—	—
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IPIUF	RCCIF	CHRIF	VMRGNIF	HMRGNIF	CMDLVIF	CMDFULIF	CMDMPTIF
bit 7							bit 0

bit 15 PUIF: бит флага завершения обработки прерывания

PUIF - комбинация из IPIUF, RCCIF и CHRIF.

1 = один или более модулей выполнил команду завершения (очищается программой)

0 = все модули в простое или заняты выполнением команды завершения

bit 7 IPIUF: бит флага состояния прерывания модуля распаковки изображения (IPU)

1 = IPU выполнил команду завершения (очищается программой)

0 = IPU состояние простоя или занят выполнением команды завершения

bit 6 RCCIF: бит флага состояния прерывания модуля обработки прямоугольников

1 = IPU выполнил команду завершения (очищается программой)

0 = IPU состояние простоя или занят выполнением команды завершения

bit 5 CHRIF: бит флага состояния прерывания модуля обработки символов

1 = IPU выполнил команду завершения (очищается программой)

0 = IPU состояние простоя или занят выполнением команды завершения

bit 4 VMRGNIF: бит флага состояния прерывания обратного хода развертки по вертикали

1 = состояние обратного хода (очищается программой)

0 = состояние прямого хода

bit 3 HMRGNIF: бит флага состояния прерывания обратного хода развертки по горизонтали

1 = состояние обратного хода (очищается программой)

0 = состояние прямого хода

bit 2 CMDLVIF: бит флага указателя числа команд в буфере FIFO

1 = достигнуто установленное число команд (очищается программой)

0 = не достигнуто установленное число команд

bit 1 CMDFULIF: бит флага заполнения буфера команд FIFO

1 = буфер FIFO заполнен (очищается программой)

0 = буфер FIFO не заполнен

PIC24FJ256DA210. Графический модуль

bit 0 CMDMPTIF: бит флага опустошения буфера команд FIFO

1 = буфер FIFO пустой (очищается программой)

0 = буфер FIFO не пустой

Регистры рабочих областей и дисплейного буфера

Регистры стартового адреса рабочих областей G1W1ADR и G1W2ADR

Рабочие области 1 и 2 предназначены для хранения видеоданных и должны начинаться с четного байта. Адреса рабочих областей могут совпадать с адресами дисплейного буфера. Все GPU используют два регистра для определения рабочей области.

G1W1ADRL – младшая часть регистра стартового адреса рабочей области 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W1ADR15	W1ADR14	W1ADR13	W1ADR12	W1ADR11	W1ADR10	W1ADR9	W1ADR8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W1ADR7	W1ADR6	W1ADR5	W1ADR4	W1ADR3	W1ADR2	W1ADR1	W1ADR0
bit 7							bit0

G1W1ADRH – старшая часть регистра стартового адреса рабочей области 1

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
-	-	-	-	-	-	-	-
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W1ADR23	W1ADR22	W1ADR21	W1ADR20	W1ADR19	W1ADR18	W1ADR17	W1ADR16
bit 7							bit0

G1W2ADRL – младшая часть регистра стартового адреса рабочей области 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W2ADR15	W2ADR14	W2ADR13	W2ADR12	W2ADR11	W2ADR10	W2ADR9	W2ADR8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W2ADR7	W2ADR6	W2ADR5	W2ADR4	W2ADR3	W2ADR2	W2ADR1	W2ADR0
bit 7							bit0

G1W2ADRH – старшая часть регистра стартового адреса рабочей области 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
-	-	-	-	-	-	-	-
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W2ADR23	W2ADR22	W2ADR21	W2ADR20	W2ADR19	W2ADR18	W2ADR17	W2ADR16
bit 7							bit0

PIC24FJ256DA210. Графический модуль

Регистры ширины и высоты рабочей области G1PUW и G1PUH

Регистры ширины и высоты рабочей области содержат размеры рабочей области дисплея в пикселях.

Регистр G1PUW

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PUW10	PUW9	PUW8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PUW7	PUW6	PUW5	PUW4	PUW3	PUW2	PUW1	PUW0
bit 7							bit0

Регистр G1PUH

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PUH10	PUH9	PUH8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PUH7	PUH6	PUH5	PUH4	PUH3	PUH2	PUH1	PUH0
bit 7							bit0

Регистры стартового адреса дисплейного буфера G1DPADR

Регистр G1DPADRL – младшая часть.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPADR15	DPADR14	DPADR13	DPADR12	DPADR11	DPADR10	DPADR9	DPADR8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPADR7	DPADR6	DPADR5	DPADR4	DPADR3	DPADR2	DPADR1	DPADR0
bit 7							bit0

Регистр G1DPADRH – старшая часть.

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPADR23	DPADR22	DPADR21	DPADR20	DPADR19	DPADR18	DPADR17	DPADR16
bit 7							bit0

Регистры ширины и высоты дисплейного буфера G1DPW и G1DPH

Регистр G1DPW

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DPW10	DPW9	DPW8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPW7	DPW6	DPW5	DPW4	DPW3	DPW2	DPW1	DP W0
bit 7							bit0

PIC24FJ256DA210. Графический модуль

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DPH10	DPH9	DPH8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
bit 7							bit0

Регистры общей ширины и высоты дисплея G1DPWT и G1DPHT

Регистры указывают ширину и высоту дисплея в пикселях. Этот параметр может быть больше чем число отображаемых пикселей, поскольку в TFT дисплеях определенное число пикселей не отображается (занято под обратный ход развертки).

Регистр ширины дисплея G1DPWT

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DPWT10	DPWT9	DPWT8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPWT7	DPWT6	DPWT5	DPWT4	DPWT3	DPWT2	DPWT1	DPWT0
bit 7							bit0

Регистр высоты дисплея G1DPHT

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DPHT10	DPHT9	DPHT8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPHT7	DPHT6	DPHT5	DPHT4	DPHT3	DPHT2	DPHT1	DPHT0
bit 7							bit0

Регистры синхронизации

Регистр активной области дисплея G1ACTDA

Регистр устанавливает число строк перед первой видимой строкой и число дисплейных тактов DISPCLK перед первым видимым пикселем.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACTLINE7	ACTLINE6	ACTLINE5	ACTLINE4	ACTLINE3	ACTLINE2	ACTLINE1	ACTLINE0
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACTPIX7	ACTPIX6	ACTPIX5	ACTPIX4	ACTPIX3	ACTPIX2	ACTPIX1	ACTPIX0
bit 7							bit0

bit 15-8 ACTLINE<7:0>: число строк перед первой активной (видимой) строкой.

Обычно, ACTLINE = VENST (G1DBLCON<15:8>).

Для режима TFT= 001 - минимальная величина равна двум.

Для режима STN mode= 010, 011, 100 - минимальная величина равна '0'.

PIC24FJ256DA210. Графический модуль

bit 7-0 ACTPIX<7:0>: число пикселей перед первым активным (видимым) пикселем (в DISPCLKs)
Обычно, ACTPIX = HENST (G1DBLCON<7:0>). Величина вычисляется в циклах тактовой частоты in DISPCLK. Она зависит от величины DPGWIDTH в регистре G1CON2<15:14>.

Регистр горизонтальной синхронизации G1HSYNC

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HSLEN7	HSLEN6	HSLEN5	HSLEN4	HSLEN3	HSLEN2	HSLEN1	HSLEN0
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HSST7	HSST6	HSST5	HSST4	HSST3	HSST2	HSST1	HSST0
bit 7							bit0

bit 15-8 HSLEN<7:0>: ширина импульса HSYNC (в DISPCLKs), минимальное значение '1'.
Бит DPHSOE (G1CON3<0>) должен быть установлен для сигнала HSYNC в состояние включено.

bit 7-0 HSST<7:0>: стартовая задержка HSYNC (в DISPCLKs).
Число циклов DISPCLK от начала обратного хода развертки до начала активной строки.

Регистр вертикальной синхронизации G1VSYNC

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VSLEN7	VSLEN6	VSLEN5	VSLEN4	VSLEN3	VSLEN2	VSLEN1	VSLEN0
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VSST7	VSST6	VSST5	VSST4	VSST3	VSST2	VSST1	VSST0
bit 7							bit0

bit 15-8 VSLEN<7:0>: ширина импульса VSYNC в строках), минимальное значение '1'.
Бит DPVSOE (G1CON3<1>) должен быть установлен для разрешения сигнала VSYNC .

bit 7-0 VSST<7:0>: задержка VSYNC (в строках)
Число строк от начала обратного хода развертки до начала активной строки.

Регистр синхронизирующего импульса G1DBLCON

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VENST7	VENST6	VENST5	VENST4	VENST3	VENST2	VENST1	VENST0
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HENST7	HENST6	HENST5	HENST4	HENST3	HENST2	HENST1	HENST0
bit 7							bit 0

bit 15-8 VENST<7:0>: число строк от начала импульса вертикальной синхронизации до первой отображаемой строки в кадре.

bit 7-0 HENST<7:0>: число циклов DISPCLK от начала импульса горизонтальной синхронизации до первого отображаемого пикселя каждой отображаемой строки.

PIC24FJ256DA210. Графический модуль

Регистр установки опережения прерывания G1MRGN

R/W-0	R/W-0	RW-0	R/W-0	R/W-0	R/W-0	R/W-0	RAIV-0
VBAMGN7	VBAMGN6	VBAMGN5	VBAMGN4	VBAMGN3	VBAMGN2	VBAMGN1	VBAMGN0
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HBAMGN7	HBAMGN6	HBAMGN5	HBAMGN4	HBAMGN3	HBAMGN2	HBAMGN1	HBAMGN0
bit 7							bit0

bit 15-8 VBAMGN<7:0>: биты, определяющие интервал до начала импульса вертикальной синхронизации. Интервал до начала импульса вертикальной синхронизации определяется в периодах DISPCLK для определения точного момента прерывания по импульсу вертикальной синхронизации VSYNC.

bit 7-0 HBAMGN<7:0>: биты, определяющие интервал до начала импульса горизонтальной синхронизации. Интервал до начала импульса горизонтальной синхронизации в периодах DISPCLK для определения точного момента прерывания по импульсу горизонтальной синхронизации HSYNC.

Регистры цветовой таблицы

Регистр записи данных в цветовую таблицу G1CLUTWR

R/W-0	R/W-0	R/W-0	RM/-0	R/W-0	R/W-0	R/W-0	RM-0
CLUTWR15	CLUTWR14	CLUTWR13	CLUTWR12	CLUTWR11	CLUTWR10	CLUTWR9	CLUTWR8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLUTWR7	CLUTWR6	CLUTWR5	CLUTWR4	CLUTWR3	CLUTWR2	CLUTWR1	CLUTWR0
bit 7							bit0

bit 15-0 CLUTWR<15:0>: биты данных, записываемые в цветовую таблицу.

Запись в регистр слова данных или старшего байта вызывает запись данных в цветовую таблицу по адресу в CLUTADR. Запись младшего байта только обновляет данные в регистре без записи в таблицу.

Регистр чтения данных из цветовой таблицы G1CLUTRD

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CLUTRD15	CLUTRD14	CLUTRD13	CLUTRD12	CLUTRD11	CLUTRD10	CLUTRD9	CLUTRD8
bit 15							bit 8
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CLUTRD7	CLUTRD6	CLUTRD5	CLUTRD4	CLUTRD3	CLUTRD2	CLUTRD1	CLUTRD0
bit 7							bit0

bit 15-0 CLUTRD<15:0>: Содержит последние данные прочитанные из цветовой таблицы по адресу (G1CLUT<7:0>). Регистр защелкивает данные по переходу из '0' в '1' бита CLUTTRD (G1CLUT<9>).

PIC24FJ256DA210. Графический модуль

Управляющий регистр цветовой таблицы G1CLUT

R/W-0	R-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
CLUTEN	CLUTBUSY	—	—	—	—	CLUTTRD	CLUTRWEN
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLUTADR7	CLUTADR6	CLUTADR5	CLUTADR4	CLUTADR3	CLUTADR2	CLUTADR1	CLUTADR0
bit 7							bit0

bit 15 CLUTEN: бит разрешения использования цветовой таблицы.

1 = цветовая таблица разрешена

0 = цветовая таблица запрещена

bit 14 CLUTBUSY: бит занятости цветовой таблицы.

1 = выполняется чтение/запись в таблице

0 = таблица свободна

bit 9 CLUTTRD: бит переключения чтения таблицы.

Переключает таблицу в режим чтения по адресам установленным в разрядах CLUTADR (G1CLUT<7:0>) с разрешением в бите CLUTRWEN.

1 = чтение CLUT разрешено (очищается программно после чтения)

0 = чтение CLUT запрещено

bit 8 CLUTRWEN: бит разрешения чтения/записи в цветовую таблицу

Бит разрешает чтение/запись таблицы со стороны программы. Должен быть очищен для доступа со стороны контроллера.

1 = чтение/запись разрешено, доступ контроллеру закрыт

0 = чтение/запись запрещено, доступ контроллеру открыт

bit 7-0 CLUTADR<7:0>: адрес цветовой таблицы

PIC24FJ256DA210. Графический модуль

Другие регистры

Регистры координат символа G1CHRX и G1CHRY

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	CURPOSX10	CURPOSX9	CURPOSX8
bit 15							bit 8
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CURPOSX7	CURPOSX6	CURPOSX5	CURPOSX4	CURPOSX3	CURPOSX2	CURPOSX1	CURPOSX0
bit 7							bit0

bit 10-0 CURPOSX<10:0>: Координата X

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	-		—	CURPOSY10	CURPOSY9	CURPOSY8
bit 15							bit 8
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CURPOSY7	CURPOSY6	CURPOSY5	CURPOSY4	CURPOSY3	CURPOSY2	CURPOSY1	CURPOSY0
bit 7							bit0

bit 10-0 CURPOSY<10:0>: Координата Y

Регистр состояния модуля распаковки изображения G1IPU

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0
—	—	HUFFERRW ⁽²⁾	BLCKERRW ⁽²⁾	LENERRW ⁽²⁾	WRAPERRW ⁽²⁾	IPUDONE ^(1,2)	BFINAL ^(1,2)
bit 7							bit0

bit 5 HUFFERR: бит состояния неопределенного кода⁽²⁾

1 = встретился неопределенный код

0 = нет неопределенного кода

bit 4 BLCKERR: бит состояния неопределенного блока кода⁽²⁾

1 = встретился неопределенный блок

0 = не встретился неопределенный блок

bit 3 LENERR: бит несоответствия длины блока⁽²⁾

1 = несоответствие длины блока

0 = нет несоответствия длины блока

bit 2 WRAPERR: бит общей ошибки⁽²⁾

1 = общая ошибка

0 = нет ошибки

bit 1 IPUDONE: бит декомпрессии IPU^(1,2)

1 = декомпрессия выполнена

0 = декомпрессия не выполнена

bit 0 BFINAL: бит состояния последнего блока^(1,2)

PIC24FJ256DA210. Графический модуль

1 = встретился последний блок

0 = не встретился последний блок

Заметка

1: Биты IPUDONE и BFINAL устанавливаются после успешной декомпрессии.

2: Все биты IPU доступны после каждого цикла декомпрессии. Все биты состояния автоматически очищаются после команды декомпрессии (IPU_DECOMPRESS).

Регистр разрешения контактов выходных данных G1DBEN

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GDBEN15	GDBEN14	GDBEN13	GDBEN12	GDBEN11	GDBEN10	GDBEN9	GDBEN8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GDBEN7	GDBEN6	GDBEN5	GDBEN4	GDBEN3	GDBEN2	GDBEN1	GDBEN0
bit 7							bit0

bit 15-0 GDBEN<15:0>: бит разрешения выходных контактов шины видеоданных.

1 = контакты отображения данных (GD<x>) разрешены

0 = контакты отображения данных (GD<x>) запрещены

GDBEN<15:0> может использоваться для управления сигналами данных, когда установлен бит DPPINOE (G1CON3<9>).

DPPINOE

GDBENx (where x = 0 to 15)

1	1	Сигналы (GD) ассоциированные с GDBENx - разрешены.
1	0	Сигналы (GD) ассоциированные с GDBENx - запрещены.
0	безразлично	Сигналы (GD) ассоциированные с GDBENx - разрешены.

PIC24FJ256DA210. Графический модуль

Сводная таблица регистров

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0	All Resets	
G1CMDL	GCMD<15:0>																0000	
G1CMDH	GCMD<31:16>																0000	
G1CON1	G1EN	—	G1SIDL	GCMDWMK<4:0>					PUBPP<2:0>			GCWDCNT<4:0>					0000	
G1STAT	PUBUSV	—	—	—	—	—	—	—	IPUBUSY	RCCBUSY	CHRBUSY	VMRGN	HMRGN	CMDLV	CMDFUL	CMDMPT	0000	
G1IE	PU IE	—	—	—	—	—	—	—	IPUIE	RCCIE	CHRIE	VMRGNIE	HMRGNIE	CMDLVIE	CMDFULIE	CMDMPTIE	0000	
G1IR	PUIF	—	—	—	—	—	—	—	IPUIF	RCCIF	CHRIF	VMRGNIF	HMRGNIF	CMDLVIF	CMDFULIF	CMDMPTIF	0000	
G1W1ADRL	W1ADR<15:0>																0000	
G1W1ADRH	—	—	—	—	—	—	—	—	W1ADR<23:16>									0000
G1W2ADRL	W2ADR<15:0>																0000	
G1W2ADRH	—	—	—	—	—	—	—	—	W2ADR<23:16>									0000
G1PUW	—	—	—	—	—	PUW<10:0>											0000	
G1PUH	—	—	—	—	—	PUH<10:0>											0000	
G1DPADRL	DPADR<15:0>																0000	
G1DPADRH	—	—	—	—	—	—	—	—	DPADR<23:16>									0000
G1DPW	—	—	—	—	—	DPW<10:0>											0000	
G1DPH	—	—	—	—	—	DPH<10:0>											0000	
G1DPWT	—	—	—	—	—	DPWT<10:0>											0000	
G1DPHT	—	—	—	—	—	DPHT<10:0>											0000	
G1CON2	DPGWDTH<1:0>		DPSTGER<1:0>		—	—	DPTEST<1:0>		DPBPP<2:0>			—	—	DPMODE<2:0>			0000	
G1CON3	—	—	—	—	—	—	DPPINOE	DPPOWER	DPCLKPOL	DPENPOL	DPVSPOL	DPHSPOL	DPPWROE	DPENOE	DPVSOE	DPHSOE	0000	
G1ACTDA	ACTLINE<7:0>								ACTPIX<7:0>								0000	
G1HSVNC	HSLEN<7:0>								HSST<7:0>								0000	
G1VSYNC	VSLEN<7:0>								VSST<7:0>								0000	
G1DBLC0N	VENST<7:0>								HENST<7:0>								0000	
G1CLUT	CLUTEN	CLUTBSY	—	—	—	—	CLUTTRD	CLUTRWEN	CLUTADR<7:0>									0000
G1CLUTWR	CLUTWR<15:0>																0000	
G1CLUTRD	CLUTRD<15:0>																0000	
G1MRGN	VBAMGN<7:0>								HBAMGN<7:0>								0000	
G1CHRX	—	—	—	—	—	CURPOSX<10:0>											0000	
G1CHRV	—	—	—	—	—	CURPOSY<10:0>											0000	
G1IPU	—	—	—	—	—	—	—	—	—	—	HUFFERR	BLCKERR	LENERR	WRAPERR	IPUDONE	BFINAL	0000	
G1DBEN	GDBEN<15:0>																0000	