|  |
| --- |
| STD: D0 = 0 - вкл. ROM, 1 - выкл. ROM; D1 = 0 - 5 цветов, 1 - 8 цветов.  MX: D0-D7 - цвет. |
| Я тут много думал по поводу окна процессора. Это которое не переключаемое ОЗУ в FFC0H-FFDFH. Ведь, при включении ПЗУ, все равно вся верхняя область доступна (с C000H в оригинале и с 8000H у нас), а окно приобретает смысл только при работе со второй страницей ОЗУ. И вот тут, увеличение окна до 224 байт (FF00H-FFDFH) вроде как позволит использовать шлюз перехода между страницами и хранить программы в ней. С другой стороны, в ней недоступен экран и существующий софт не будет использовать эту особенность. К тому же, для дешифрации портов, все равно надо выделять все биты адреса  ...  Карта памяти уже обсуждалась мной ранее, я лишь напомню ее области:   [Режим 0: MX RAM (запись в FFFCH)]  0000H - FFDFH = Основное ОЗУ.  FFE0H - FFFFH = Порты ввода-вывода   [Режим 1: MX EXT RAM (запись в FFFDH)]  0000H - FFBFH = Дополнительное ОЗУ  FFC0H - FFDFH = Основное ОЗУ (шлюз)  FFE0H - FFFFH = Порты ввода-вывода   [Режим 2: MX ROM (запись в FFFEH)]  0000H - 7FFFH = ПЗУ МХа, включая систему в первых 16КБ  8000H - FFDFH = Основное ОЗУ  FFE0H - FFFFH = Порты ввода-вывода   [Режим 3: STD (запись в FFFFH или по сбросу)]  0000H - 7FFFH = ПЗУ (секция STD) или основное ОЗУ, управление: FFFAH  8000H - EFFFH = Основное ОЗУ  F000H - FFFFH = Порты ввода-вывода, причем в области F800H-FFFFH блокируется на ППА клавиатуры   Все просто, запуск уже обсуждали. Теперь о страницах ПЗУ и доп. ОЗУ. Я тоже много думал - они не нужны. У нас есть карта памяти. И быстрая загрузка/выгрузка с/на нее. Это упрощает шлюз (который так и сделан у RAMFOS'а): вне зависимости от значения [A] они пишут в нужную ячейку порта для переключения режима, поэтому сам шлюз получается предельно простым:   |  | | --- | | **Код:** | | READ:  STA 0FFFDH  MOV  A,M  STA 0FFFCH  RET  WRITE:  STA 0FFFDH  MOV  M,A  STA 0FFFCH  RET |   Вполне умещаемся в 32 байта. И вообще, в силу определенных обстоятельств, я предлагаю вторую страницу ОЗУ использовать только под данные. Например, программа работает в основном ОЗУ, а дополнительное - чисто ее данные (текст, для текстового реактора и т.д.). Или "кармана" для программ, работающих с данными в основном ОЗУ. Так что, страницы не нужны. Окончательное решение. |
| Так вот, карта памяти следующая. По сбросу включен **Mode\_3**, который на самом деле является стандартным специалистом. При этом, порт клавиатуры расположен по адресам F800-FFFF, весь набор портов по адресам F000-F7FF, 8 устройств по 4 ячейки. При этом, клавиатурный ППА доступен так же по адреса F000-F003 и так далее до F7E0..F7E3. Для аналогии с режимом МХа, предлагаю использовать именно F7E0..F7FF для обращения к портам. Младшие 32КБ занимает ПЗУ, которое может быть отключено (по сбросу оно включается автоматически). Остальную область занимает основное ОЗУ с экраном. ПЗУ в **Mode\_3** может быть отключено специальным регистром. Я его расположил рядом с регистром цвета. Т.е., F7F8-F7F9 - это регистр цвета, а F7FA-F7FB - регистр включения ПЗУ в режиме **Mode\_3**. Сам регистр режимов доступен по адресам F7FC-F7FF в режиме **Mode\_3** и FFFC-FFFF в остальных 3х (**Mode\_0**, **Mode\_1** и **Mode\_2** - это режимы МХа). При этом, **Mode\_0** соответствует выключению ПЗУ и RAM диска в режиме МХ, **Mode\_1** соответствет включению RAM диска в режиме МХ (при этом я пока сохранил дешифрацию МХа: диск занимает с 0000 по FFBF, в своем скорее всего порежу диск до 0000-FEFF, что упростит схему дешифратора), а **Mode\_2** соответствет включению внутреннего ПЗУ в режиме МХ. Замечу, что ПЗУ стоит на 64КБ (можно и больше, но уже странично), а в карте процессора он занимает всего 32КБ. Т.е., в ПЗУ 2 страницы по 32КБ и каждая из них соответствет своему режиму (т.е., для стандартного это 0000-7FFF в ПЗУ, а для МХа - 8000-FFFF), что позволяет записывать уникальный софт для каждого режима.  Процесс загрузки я уже оглашал: по сбросу режим стандартного спеца, там пускается софт выбора режима, по типу спектрума128, если выбирается стандартный - то грузится система с карты (или ПЗУ, если карта не вставлена, круто же?) и ПЗУ отключается. Либо формируется шлюз в окне процессора (FFC0-FFDF), который переключает режим на **Mode\_2** и прыгает в 0000, запуская ПЗУ в режиме МХа. Как видно, все просто и логично. Думаю, карту памяти все представили и рисовать ее нет необходимости. С портами тоже все прозрачно. Обсуждаем. |