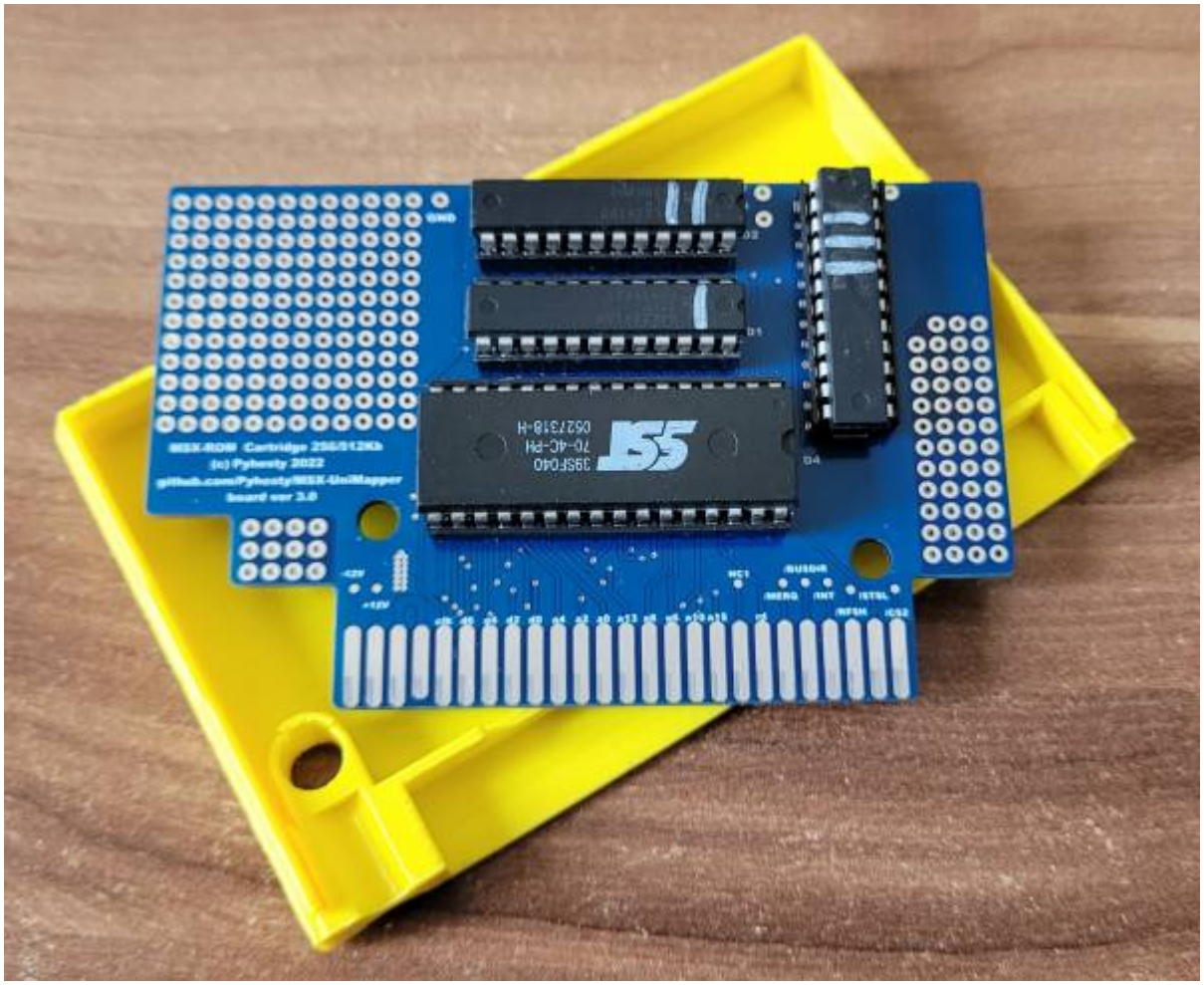
MSX UniMapper



UniMapper MSX (MSX universal mapper cartridge) для стандартов мапперов ROM:

- Konami
- Konami SCC
- ASCII8
- ASCII16

(C) Pyhesty 2021–2022, 2022 RBSC

Выполнен в виде платы для картриджа.

MSX-UniMapper

Необходимость разработки универсального маппера для наиболее распространённых мапперов компьютера MSX возникла по причине того, что имеющиеся схемы мапперов имели следующие недостатки:

1. не обеспечивали сброса состояния маппера по сигналу Reset;
2. не обеспечивали универсальности и были не взаимозаменяемые;
3. содержали достаточно большое количество корпусов, сложный или дорогой монтаж;
4. содержали откровенные ошибки в схемах, что усложняло бы их использование любителями.

С целью устранения данных недостатков было принято решение разработать маппер на PLD-микросхемах малой степени интеграции типа GAL22V10D, которые достаточно хорошо распространены, недорогие и обладают нужным функционалом. В частности:

1. 11 входов;
2. 10 выходов;
3. позволяют реализовать любую функцию входных сигналов;

4. позволяют использовать данные для функции выходных сигналов;
5. содержат 10 регистров, что позволяет хранить до 10 бит данных.

Функция маппера в картриджах MSX состоит в том, чтобы обеспечить адресацию относительно большой ПЗУ микросхемы объёмом 128кБ и более в ограниченное адресное пространство шины (все адресное пространство которой 64кБ). В стандарте MSX адресное пространство в 64кБ может быть поделено по 16кБ между разными устройствами, находящимися в так называемых слотах (подробнее здесь). Картриджу по умолчанию выдаётся слот в 32кБ начиная с адреса 0x4000 по адрес 0xBFFF. Далее картридж должен сам определять использование данного пространства. Если картридж небольшой и его ПЗУ до 64кБ, он может не применять маппер, но в случае, когда требуется больший объем, то он вынужден в каком-то из адресных пространств менять страницы ПЗУ. Как именно будет происходить смена сегментов ПЗУ — определяет на аппаратном уровне разработчик картриджа. Наиболее распространено разделение 32кБ на четыре страницы по 8кБ, это очень удобно для игр, так как для разных сцен можно хранить разные наборы данных сцены и переключать их по необходимости. При обращении по определённому адресному пространству (в общем случае одному из четырёх) должен подставляться адрес нужной страницы. Для объёма картриджа в 128кБ требуется 16 сегментов по 8кБ. Для адресации 8кБ требуется 13 бит адреса (2^{13}), эти адреса подключаются к микросхеме памяти напрямую, а старшие четыре бита адреса (16 страниц требуют 4 бита адресации) подключаются через схему маппера.

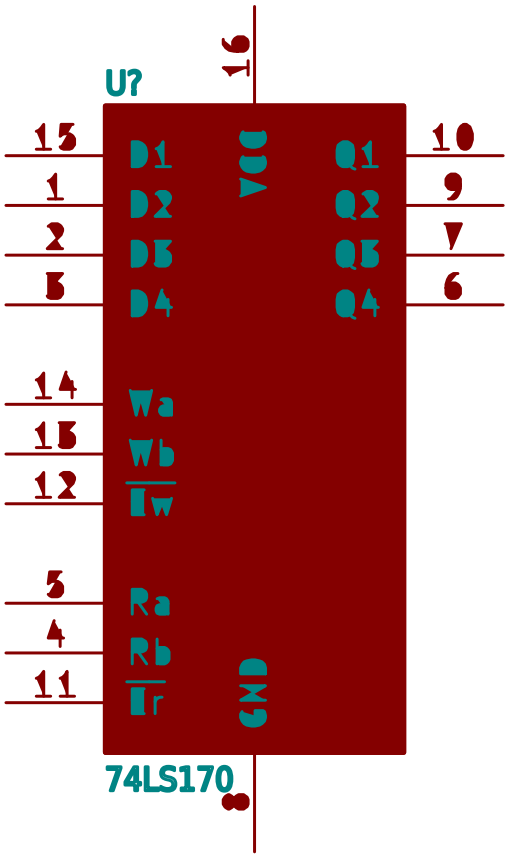
Теория

Маппер должен обеспечивать:

1. дешифрацию адреса переключения номера страницы;
2. сохранение нового номера сегменты соответствующей страницы;
3. дешифрацию адреса страницы и выдачу номера установленного сегмента.

Классически сохранением адресов сегментов занимаются микросхемы файловых регистров, такие как 74LS670 или 555ИР26, которые на самом деле представляют собой оперативное запоминающее устройство (память) на 16 бит с организацией 4 бита данных на 2 бита адреса. Это позволяет одной микросхеме хранить данные о шестнадцати сегментах для каждой из четырёх страниц. Специальная схема упрощённой дешифрации адреса устанавливает адрес на чтение и команду на запись. К сожалению, недостатком схемы, собранной на файловом регистре, будет невозможность изначальной инициализации регистров файлового регистра, что не позволит запускать некоторые ROM.

Условное графическое обозначение К555ИР26, КМ555ИР26



Назначение выводов:

№	Обозначение	Описание
15	D1	информационные входы
1	D2	
2	D3	
3	D4	
14	Wa	выборка записи
13	Wb	
12	/Ew	разрешение записи
5	Ra	выборка чтения
4	Rb	
11	/Er	разрешение чтения
10	Q1	выходы
9	Q2	
7	Q3	
6	Q4	
8	GND	общий
16	VCC	напряжение питания

Таблицы истинности

Функции записи:

Адрес записи			Регистр			
Wb	**Wa**	**/Ew**	**0**	**1**	**2**	**3**
0	0	0	$Q_i=D_i$	Q_n	Q_n	Q_n
0	1	0	Q_n	$Q_i=D_i$	Q_n	Q_n
1	0	0	Q_n	Q_n	$Q_i=D_i$	Q_n
1	1	0	Q_n	Q_n	Q_n	$Q_i=D_i$
X	X	1	Q_n	Q_n	Q_n	Q_n

Функции чтения:

Адрес чтения			Выходы			
Rb	**Ra**	**/Rd**	**1**	**2**	**3**	**4**
0	0	0	W_{0B1}	W_{0B2}	W_{0B3}	W_{0B4}
1	0	0	W_{1B1}	W_{1B2}	W_{1B3}	W_{1B4}
1	0	0	W_{2B1}	W_{2B2}	W_{2B3}	W_{2B4}
1	1	0	W_{3B1}	W_{3B2}	W_{3B3}	W_{3B4}
X	X	1	Z			

Примечание: W_{iBj} — регистр i, разряд j.

Описание КМ555ИР26

Konami

64 страницы × 8 кБайт, объём до 512 кБайт

Маппер является наиболее простым. Нулевая страница всегда содержит нулевой сегмент. Дополнительным является требование, что первая страница должна содержать первый сегмент. Это невозможно реализовать на простом маппере с файловым регистром без применения сложный схемы инициализации страниц по сигналу Reset. К счастью большинство игр самостоятельно выполняют инициализацию первой страницы. Количество бит для реализации маппера: 3 * $\log_2$(количество сегментов). В частности, для 128кБ требуется 3*4 = 12 бит, для 256кБ = 16 бит и для 512кБ = 20 бит.

Страница (8кБ)	Адрес переключения	Начальный сегмент
0x4000~0x5FFF (зеркало: 0xC000~0xDFFF)	—	Всегда 0
0x6000~0x7FFF (зеркало: 0xE000~0xFFFF)	0x6000 (зеркало: 0x6001~0x7FFF)	1
0x8000~0x9FFF (зеркало: 0x0000~0x1FFF)	0x8000 (зеркало: 0x8001~0x9FFF)	Псевдослучайный (random)
0xA000~0xBFFF (зеркало: 0x2000~0x3FFF)	0xA000 (зеркало: 0xA001~0xBFFF)	

Konami's MegaROMs without SCC

ASCII8

32 страницы × 8 кБайт, объём до 256 кБайт

Маппер аналогичен Konami, но,

- во-первых, значительно упрощена инициализация сегментов — они все должны быть равны нулю (к сожалению, некоторые ROM требуют уникальную инициализацию сегментов),

- во-вторых, возможно переключение нулевого сегмента,
- в-третьих, отличные адреса переключения от Konami

Количество требуемых бит для хранения: 4 * $\log_2$(количество сегментов). В частности, для 128кБ требуется 4*4 = 16 бит, для 256кБ = 20 бит и для 512кБ = 24 бита. К счастью, большинство ROM не требует переключение сегментов нулевой страницы.

Страница (8кБ)	Адрес переключения	Начальный сегмент
0x4000~0x5FFF (зеркало: 0xC000~0xDFFF)	0x6000 (зеркало: 0x6001~0x67FF)	0
0x6000~0x7FFF (зеркало: 0xE000~0xFFFF)	0x6800 (зеркало: 0x6801~0x68FF)	
0x8000~0x9FFF (зеркало: 0x0000~0x1FFF)	0x7000 (зеркало: 0x7001~0x77FF)	
0xA000~0xBFFF (зеркало: 0x2000~0x3FFF)	0x7800 (зеркало: 0x7801~0x7FFF)	

Примечание: Зеркала страниц отсутствуют на многих картриджах.

ASCII8

ASCII16

32 страницы × 16 кБайт, объём до 512 кБайт

Маппер имеет следующие отличия:

1. размер страницы: 16кБ,
2. количество страниц: две.

Количество требуемых бит для хранения: 2 * $\log_2$(количество сегментов). В частности, для 128кБ требуется 2*4 = 8 бит, для 256кБ = 10 бит и для 512кБ = 12 бит. К счастью, большинство ROM не требует переключение сегментов нулевой страницы.

Страница (16кБ)	Адрес переключения	Начальный сегмент
0x4000~0x7FFF (зеркало: 0xC000~0xFFFF)	0x6000 (зеркало: 0x6001h~0x7FFF)	0 0x0F для R-Type (*)
0x8000~0xBFFF (зеркало: 0x0000~0x7FFF)	0x7000 (зеркало: 0x7001~0xBFFF)	0

Примечания:

- Зеркала страниц отсутствуют на многих картриджах.
-

  R-Type — 384 кБ использует тот же маппер, но сегмент 0x0F остаётся фиксированным на странице 0x4000~0x7FFF. Кроме того, сегмент 0x0F такой же, как 0x17 (последний сегмент).

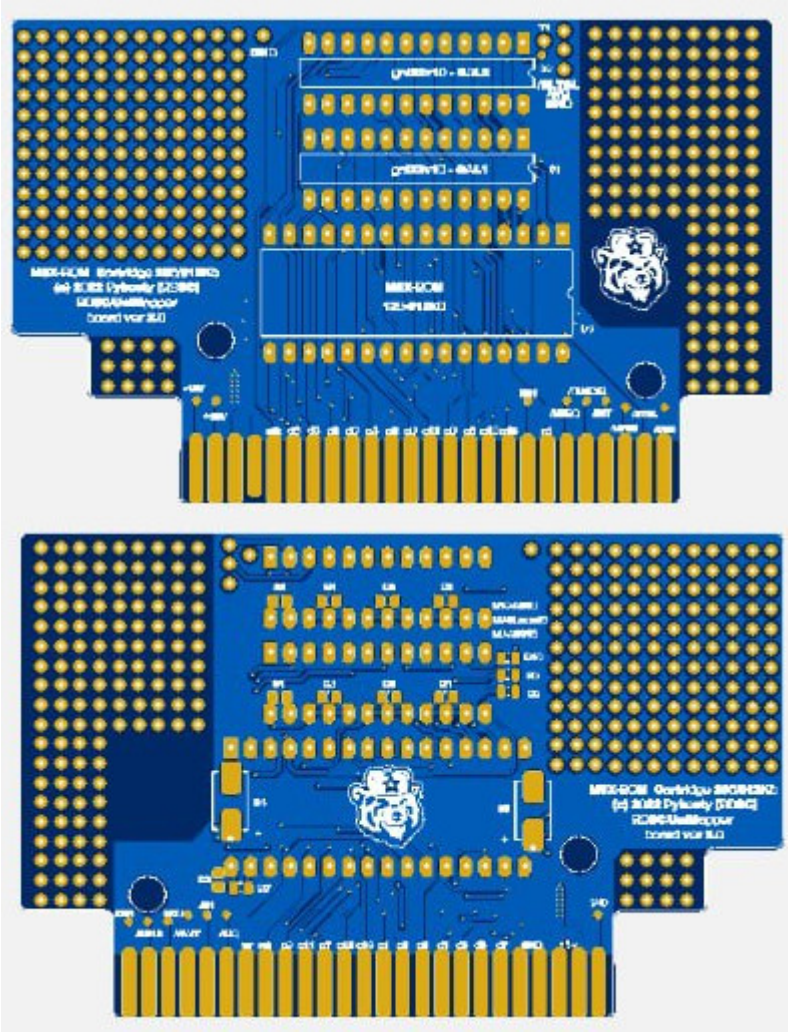
ASCII16

Konami SCC

Страница (16кБ)	Адрес переключения	Начальный сегмент
0x4000~0x5FFF (зеркало: 0xC000~0xDFFF)	0x5000 (зеркало: 0x5001~0x57FF	0
0x6000~0x7FFF (зеркало: 0xE000~0xFFFF)	0x7000 (зеркало: 0x7001~0x77FF)	1
0x8000~0x9FFF (зеркало: 0x0000~0x1FFF)	0x9000 (зеркало: 0x9001~0x97FF)	2
0xA000~0xBFFF (зеркало: 0x2000~0x3FFF)	0xB000 (зеркало: 0xB001~0xB7FF)	3

Konami's MegaROMs with SCC

Версия 2.0



Firmware v2.0

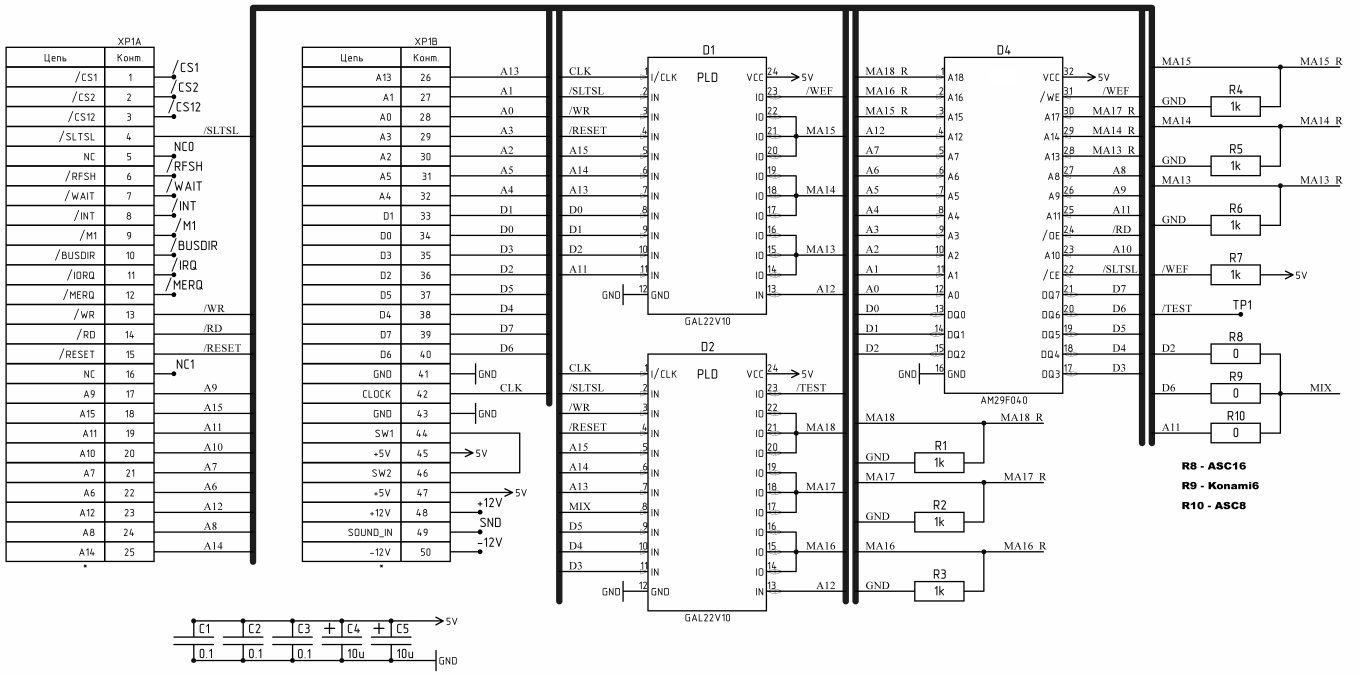
Версия 1.0 была максимально гибкой, и содержала излишнее количество элементов, в данном описании не упоминается, хотя присутствует в репозитории.

Схема 2.0 максимально упрощена, предполагается, что 0-я страница всегда содержит нулевой сегмент. Это условие выполняется для большинства известных ROM и всегда выполняется для стандарта Konami. Адреса A0 – A12 заведены непосредственно на ПЗУ.

Адреса A13 – A18 поступают с маппера, которые реализован на GAL22V10

Сигнал выбора слота /SLTSL, записи /WR и сброса /RESET заведены на комбинационные входы PLD (GAL22V10), так же заведена тактовая частота, которая требуется для синхронной работы внутренних регистров.

Объединение выводов PLD 14 – 16, 17 – 19, 20 – 22 обеспечивают монтажное ИЛИ выходных регистров PLD. Резисторы R1 – R6 — обеспечивают подтяжку выводов к «0», когда выходы PLD выключены и неактивны, что обеспечивает ещё одно (четвертое) состояние выхода, при выборе нулевой страницы.



оригинал

Konami

Описание работы маппера в режиме Konami без SCC

PLD содержит 10 регистров (или бит памяти) и позволяет реализовать до 10 комбинационных схем. Каждая схема каждого бита содержит: предустановку бита, установку бита и сохранение этого бита во всех остальных случаях.

Вот пример описания всех четырёх состояний одного бита. Условно первое число отвечает за номер бита, второе за номер страницы (которых как раз четыре, как и количество состояний бита). В зависимости от установленного адреса активируется ответственный за текущее состояние регистр.

```
d00.d  – на схеме через резистор всегда подтянут к "0"
d01.d = !nReset#(((!nWE&!nSTSL)&A13&!A15&di0)#((nWE#nSTSL)&d01)#((!nWE&!nSTSL)&(!A13#A15)&d01));
d02.d =  nReset&(((!nWE&!nSTSL)&!A13&A15&di0)#((nWE#nSTSL)&d02)#((!nWE&!nSTSL)&(A13#!A15)&d02));
d03.d = !nReset#(((!nWE&!nSTSL)&A13&A15&di0)#((nWE#nSTSL)&d03)#((!nWE&!nSTSL)&(!A13#!A15)&d03));
```

Разберём более подробно регистр d01, который отвечает за состояние бита при адресации первой страницы:

```
d01.d = !nReset#(((!nWE&!nSTSL)&A13&!A15&di0)#((nWE#nSTSL)&d01)#((!nWE&!nSTSL)&(!A13#A15)&d01));
```

- `!nReset#`

  отвечает за предустановку регистра при сигнале сброс в «1», таком образом нулевой бит для первой страницы будет равен «1», что соответствует стандарту.

- `(!nWE&!nSTSL)`

  дешифратор сигнала записи, когда выбран слот `/STSL` и поступает сигнал `/WE`, значит происходит запись в адресное пространство слота.

- `A13&!A15`

  дешифратор адреса первой страницы, для дешифрации достаточно адресных битов A15 и A13

- `((!nWE&!nSTSL)&A13&!A15&di0)`

в случае если дешифрован адрес и пришёл сигнал записи, то бит устанавливается в соответствии с битом, установленным на шине данных, `di0`

- `((nWE#nSTSL)&d01)`

  в случае, если не пришёл сигнал записи, то состоянии бита сохраняется тем, что хранится в регистре `d01`

- `((!nWE&!nSTSL)&(!A13#A15)&d01))`

  и наконец, последний возможный вариант, поступили сигналы записи по адресу слота, но указан адрес другой страницы, то состояние бита сохраняется (`d01`).

Аналогичная логика применена и для всех остальных битов. Для чтения данных состояния битов, задающих сегменты для каждой из страниц, применено монтажное ИЛИ, то есть вне микросхемы PLD сигналы объединены и допускается активность только одного из выходов/регистров состояния для каждого из битов.

Их мультиплицирование производится включением и выключением выходов регистров. Когда выход выключен, он находится в Z-состоянии и не влияет на состояние выходов. Когда все выходы выключены, то работает подтяжка (тот самым монтажный d00) и состояние бита равно 0, что соответствует нулевому сегменту первой страницы.

Вот так происходит включение регистров в зависимости от адреса страниц (сигналы адреса A13 и A15 задают номер страницы)

```
d01.oe = A13&!A15; — выбор первой страницы
d02.oe = !A13&A15; — выбор второй страницы
d03.oe = A13&A15;  — выбор третьей страницы
d11.oe = A13&!A15; — и т. д.
d12.oe = !A13&A15;
d13.oe = A13&A15;
d21.oe = A13&!A15;
d22.oe = !A13&A15;
d23.oe = A13&A15;
```

Таким образом 9 битов PLD сохраняют состояние 3 битов адреса сегмента для всех четырёх страниц.

Чтобы подключить ещё 3 бита адреса сегмента (чтобы расширить количество сегментов до $2^6 = 64$), требуется ещё одна PLD.

ASCII8

Описание работы маппера в режиме

Маппер для ASCII8 аналогичен мапперу для Konami, но требуется большее количество бит для дешифровки адреса устанавливаемого сегмента (так как отличается адресация). Также часть функции, отвечающая за сброс по сигналу Reset, одинаковая для всех бит, и сбрасывает их в ноль.

```
/* 1 - page bit 0 */
d01.d=nReset&
((((!nWE&!nSTSL)&!A15&A14&!A12&A11&di0)#((nWE#nSTSL)&d01)#((!nWE&!nSTSL)&(A15#!A14#A12#!A11)&
d01));

/* 2 - page bit 0 */
d02.d=nReset&
((((!nWE&!nSTSL)&!A15&A14&A12&!A11&di0)#((nWE#nSTSL)&d02)#((!nWE&!nSTSL)&(A15#!A14#!A12#A11)&
d02));

/* 3 - page bit 0 */
d03.d=nReset&
(((!nWE&!nSTSL)&!A15&A14&A12&A11&di0)#
((nWE#nSTSL)&d03)#((!nWE&!nSTSL)&(A15#!A14#!A12#!A11)&d03));
```

ASCII16

Описание работы маппера в режиме ASCII16

Маппер для ASCII16 отличается, для хранения состояния бита сегмента требуется не четыре бита, а только два, поэтому часть битов просто не используется. Нулевой бит адреса сегмента пробрасывает адрес A13. Аналогично Konami и ASCII8 реализован сброс, установка и чтение.

```
/* 0 - page bit 0 */
d00   = A13;

/* 0 - page bit 1 */
d10.d = nReset&
((((!nWE&!nSTSL)&!A15&A14&A13&!A12&di0)#((nWE#nSTSL)&d10)#((!nWE&!nSTSL)&(A12#!A13#!A14#A15)&d10)
);

/* 1 - page bit 1 */
d11.d = nReset&((((!nWE&!nSTSL)&!A15&A14&A13&A12&di0)#
((nWE#nSTSL)&d11)#((!nWE&!nSTSL)&(!A12#!A13#!A14#A15)&d11));

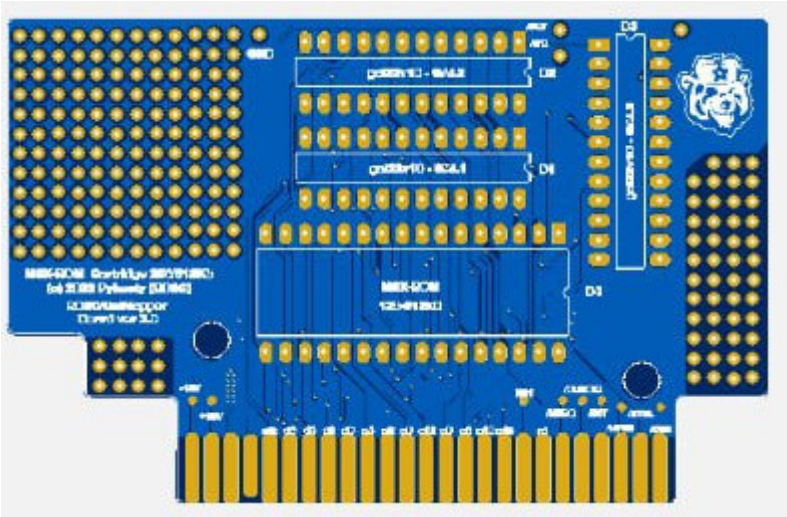
/* unuse */
d19.d = 'b'0;

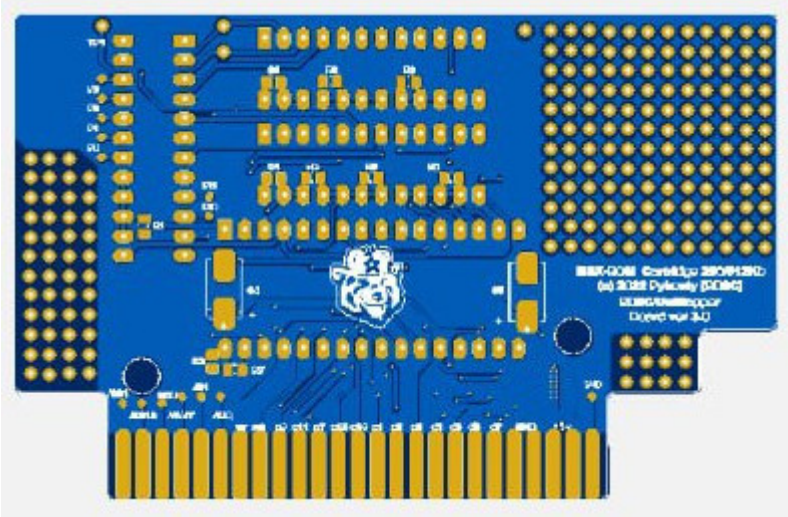
/* 0 - page bit 2 */
d20.d = nReset&
((((!nWE&!nSTSL)&!A15&A14&A13&!A12&di1)#((nWE#nSTSL)&d20)#((!nWE&!nSTSL)&(A12#!A13#!A14#A15)&d20)
);

/* 1 - page bit 2 */
d21.d = nReset&
((((!nWE&!nSTSL)&!A15&A14&A13&A12&di1)#((nWE#nSTSL)&d21)#((!nWE&!nSTSL)&(!A12#!A13#!A14#A15)&d21)
);

/* unuse */
d29.d = 'b'0;
```

Версия 3.0

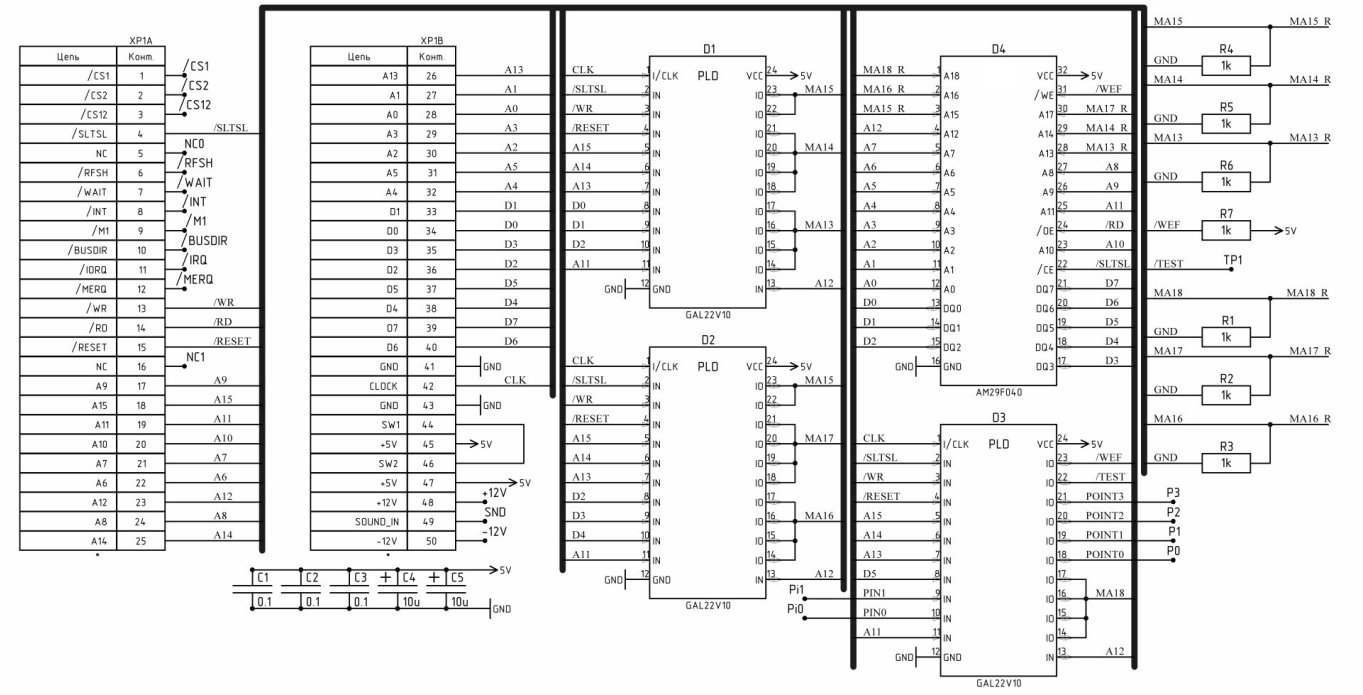




Версия 3.0 позволяет за счёт добавления ещё одной PLD реализовать Konami SCC-маппер, а также полностью совместимый маппер ASCII8 и аналогичные, объёмом до 512кБ, а также без применений дополнительной PLD-микросхемы (без D3) маппер объёмом до 256кБ. Это может быть особенно важно при использовании ROM, генерированной с применением программы DSK2ROM, которая преобразует образ диска в образ ПЗУ.

Теперь каждая GAL на выходе реализует монтажное ИЛИ для 2½ бит адреса сегмента. Для каждой из страниц реализуются все возможные состояния адресов сегмента. GAL1 и GAL2 выполнены фактически симметрично, GAL3 (D3) использует только четыре регистра.

Количество пассивных элементов уменьшено, фактически резисторы R2 – R6 не требуются, так как теперь состояния всех выходов адреса сегмента полностью определены в каждый момент времени.



оригинал

Список компонентов

Обозначение	Номинал/марка	Количество	Примечание
C1, C2, C3	0.1 мкФ	3	типоразмер 0603
C4, C5	10 мкФ	2	
R1–R7	1 кОм	7[1)]	типоразмер 0603
D1, D2, D3	GAL22V10D	3	корпус DIP-24 (узкий)

Обозначение	Номинал/марка	Количество	Примечание
D4	SST39SF020 или SST39SF040	1	корпус DIP-32

Konami SCC

Описание маппера Konami SCC

Структура регистра аналогичная, только вместо монтажного 0, задающего адрес нулевого сегмента (подтягивающего резистора) используется явный регистр d00 (и аналогичные d10, d20 и тд). Это необходимо, чтобы обеспечить адресацию в нулевой странице всех сегментов ПЗУ.

```
/* 0x5000 0b0101 */
d00.d = nReset&
((((!nWE&!nSTSL)&A12&!A13&A14&!A15&di0)#((nWE#nSTSL)&d00)#((!nWE&!nSTSL)&(!A12#A13#!A14#
A15)&d00));

/* 0x7000 0b0111 */
d01.d = nReset&
((((!nWE&!nSTSL)&A12&A13&A14&!A15&di0)#((nWE#nSTSL)&d01)#((!nWE&!nSTSL)&(!A12#!A13#!A14#
A15)&d01));

/* 0x9000 0b1001 */
d02.d = nReset&
((((!nWE&!nSTSL)&A12&!A13&!A14&A15&di0)#((nWE#nSTSL)&d02)#((!nWE&!nSTSL)&(!A12#A13#
A14#!A15)&d02));

/* 0xB000 0b1101 */
d03.d = nReset&
((((!nWE&!nSTSL)&A12&A13&!A14&A15&di0)#((nWE#nSTSL)&d03)#((!nWE&!nSTSL)&(!A12#!A13#
A14#!A15)&d03));

/* 0x5000 0b0101 */
d10.d = nReset&
((((!nWE&!nSTSL)&A12&!A13&A14&!A15&di1)#((nWE#nSTSL)&d10)#((!nWE&!nSTSL)&(!A12#A13#!A14#
A15)&d10));

/* 0x7000 0b0111 */
d11.d = nReset&
((((!nWE&!nSTSL)&A12&A13&A14&!A15&di1)#((nWE#nSTSL)&d11)#((!nWE&!nSTSL)&(!A12#!A13#!A14#
A15)&d11));

/* 0x9000 0b1001 */
d12.d = nReset&
((((!nWE&!nSTSL)&A12&!A13&!A14&A15&di1)#((nWE#nSTSL)&d12)#((!nWE&!nSTSL)&(!A12#A13#
A14#!A15)&d12));

/* 0xB000 0b1101 */
d13.d = nReset&
((((!nWE&!nSTSL)&A12&A13&!A14&A15&di1)#((nWE#nSTSL)&d13)#((!nWE&!nSTSL)&(!A12#!A13#
A14#!A15)&d13));

/* 0x5000 0b0101 */
d20.d = nReset&
((((!nWE&!nSTSL)&A12&!A13&A14&!A15&di2)#((nWE#nSTSL)&d20)#((!nWE&!nSTSL)&(!A12#A13#!A14#
A15)&d20));

/* 0x7000 0b0111 */
d21.d = nReset&
((((!nWE&!nSTSL)&A12&A13&A14&!A15&di2)#((nWE#nSTSL)&d21)#((!nWE&!nSTSL)&(!A12#!A13#!A14#
A15)&d21));
```

Аналогично, для всех регистров определены все их состояния выходов в зависимости от адреса страницы.

```
d00.oe = !A13& !A15;
d01.oe =  A13& !A15;
d02.oe = !A13&  A15;
d03.oe =  A13&  A15;
d10.oe = !A13& !A15;
d11.oe =  A13& !A15;
d12.oe = !A13&  A15;
d13.oe =  A13&  A15;
d20.oe = !A13& !A15;
d21.oe =  A13& !A15;
```

DSK2ROM

Fix Me!

DSK2ROM

Запись ППЗУ с MSX

Fix Me!

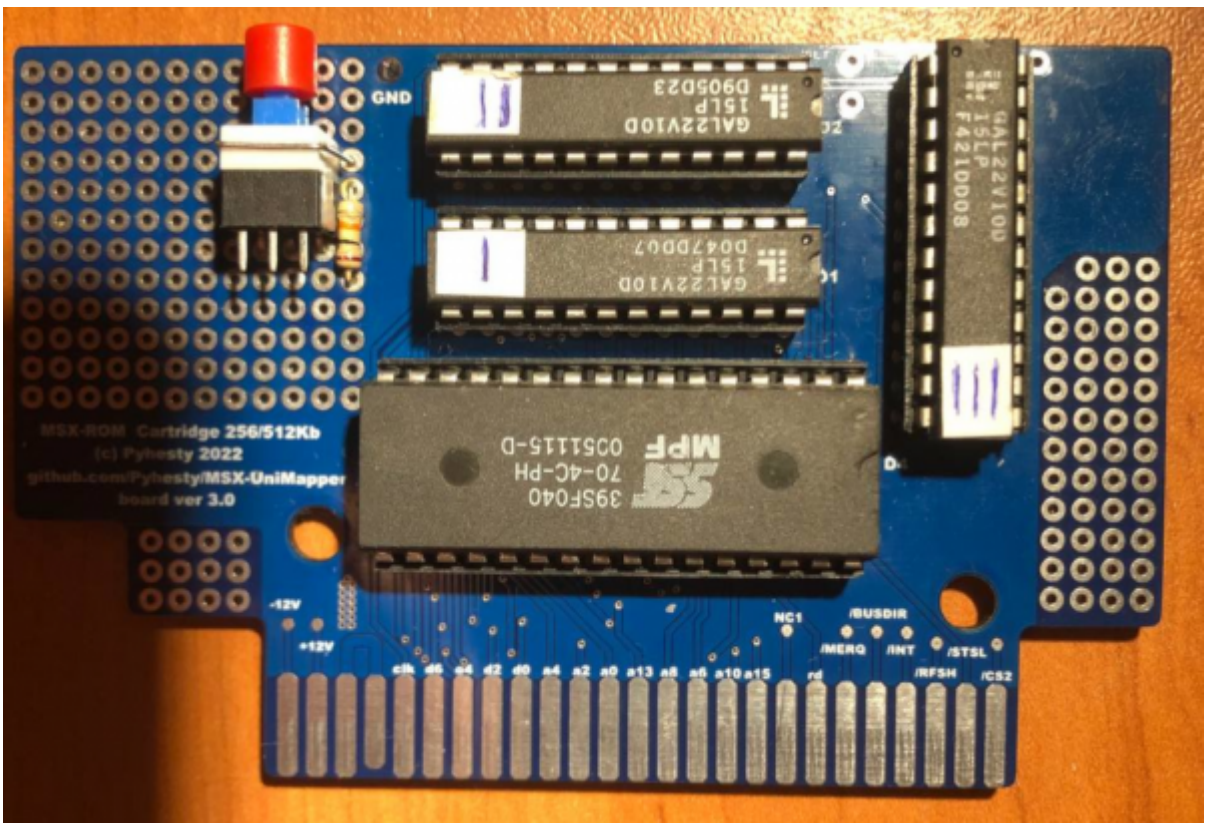


Ограничения текущей версии платы и программы:

1. поддерживается только тип микросхемы памяти SST39SF040;
2. программа записи работает только со слотом B (2), поэтому для работы в режиме записи нужно устанавливать картридж в этот слот.

Необходимые доработки

1. Установить переключатель режима работы



2. Перерезать сигнал A14 в районе краевого слота



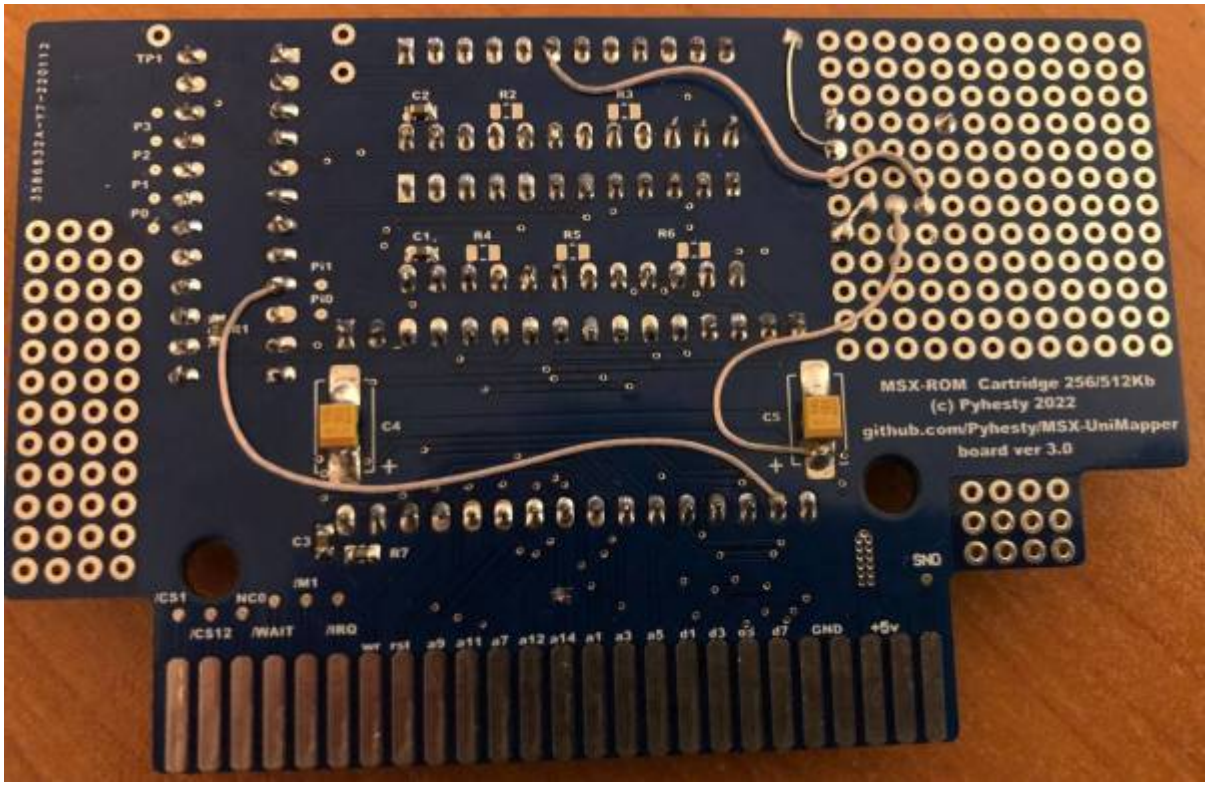
3. Выполнить подключение переключателя по схеме:

Контакт	Описание
Управление	Подключить к контакту 6 любой микросхемы GAL
Режим «чтение»	Подключить к «земле» через резистор 1–10кОм
Режим «запись»	Подключить к питанию (+5В)

Высокий уровень (5В) на контакте 6 микросхемы GAL меняет режим работы схемы с «чтения» на «запись». В режиме «запись» невозможна загрузка с картриджа!

В качестве переключателя можно использовать:
PB22E08 — кнопка с фиксацией 8×8×13 + Колпачок для кнопок A28 Red, K243-25

4. Соединить контакт 9 микросхемы GAL (D3) с контактом 18 микросхемы ППЗУ (D4):



Использование

Архив с необходимыми файлами

Порядок подготовки маппера и ПО

1. Запрограммировать GAL микросхемы соответствующим маппером.
2. Записать нужные для программирования ROM на диск. Расширение файлов ROM должны быть `.rom`.
3. Записать программу `um.com`.

Порядок работы

1. Установить переключатель в режим «запись».
2. Запустить MSX и загрузить MSX-DOS 2 или Nextor.
3. Запустить программу `um.com`.
    1. Выбрать тип маппера.
    2. Вести имя ROM-файла без расширения.
    3. Дождаться результатов записи.

Что можно проконтролировать в программе записи

1. После ввода имени файла происходит поиск ПЗУ.
   Код SST39SF040: BF B7.
2. Выводится отладочная информация по текущим сегментам в страницах памяти (первые четыре байта).
3. После стирания ПЗУ во всех ячейках памяти всех сегментов код FF.

Ссылки

simple Universal MegaROM Mapper for Konami/ASC8/ASC16 (Unimapper)

MSX Универсальный маппер до 512кБ MSX-UniMapper

1)
для работы достаточно установить только резисторы R1 и R7

https://sysadminmosaic.ru/msx/unimapper/unimapper

2023-08-06 10:20

