

# Faszination Modultechnik: Nachbau des Bounty Bob Strikes Back Modules.

Eines der aufwendigsten und kompliziertesten Spiel Module, die je in Serie gebaut wurden, ist "Bounty Bob Strikes Back". Da man es gebraucht nicht unter 50 Euro bekommt, kam mir die Idee zum Nachbau. Nach einigen Fehlschlägen in der GAL-Logik gelang mir dann doch der Durchbruch. Um eine lauffähige Kopie zu erstellen muss zuerst die Arbeitsweise richtig verstanden werden.

Im Inneren des Originals befinden sich 3 Rombausteine mit folgender Adressenaufteilung:

IC1 belegt den Speicherbereich von \$8000-8FFF

IC2 belegt den Speicherbereich von \$9000-9FFF

IC3 belegt den Speicherbereich von \$A000-BFFF

IC3 ist ein normaler Rombaustein mit 8k Inhalt. IC1 und IC2 sind Sonderbausteine und besitzen intern jeweils 16k können jedoch nur 4k im Speicherbereich einblenden.

Um auf einen anderen Bereich umzuschalten sind folgende Adressen dafür reserviert:

IC1: \$8FF6 = Bank 1, \$8FF7 - Bank 2, \$8FF8 - Bank 3, \$8FF9 - Bank 4.

IC2: \$9FF6 = Bank 1, \$9FF7 - Bank 2, \$9FF8 - Bank 3, \$9FF9 - Bank 4.

Die Adressen werden dabei nur gelesen und nicht beschrieben.

Bank 1 entspricht den Bereich \$0-\$FFF des 16k Roms, Bank 2 \$1000-\$1FFF, Bank 3 \$2000-2FFF und Bank 4 \$3000-3FFF.

Der größte Aufwand liegt in der Adressdekodierung für die Bankumschaltung. Es werden alle Adressleitungen von A0 bis A12 sowie die Chipselect S4 (\$8000-\$9FFF) und S5 (\$A000-BFFF) benötigt. Bei meiner ersten Logik hatte ich alle auf ein GAL geführt.

Resultat war die Fehlermeldung: "Es steht nicht genügend Speicher für die Logik zur Verfügung" weshalb ich auf ein zweites GAL zurückgreifen mußte. GAL-Ausgang A86 und A96 werden nur zur Rücksetzung auf Bank 1 benötigt und im zweiten GAL nicht weiterverarbeitet. Sind A87,A88,A89,A97,A98,A99 nicht aktiv wird automatisch Bank 1 angewählt.

Als Beispiel soll dieser Ausdruck des ersten GAL's erläutert werden:

A87 = A0 & A1 & A2 & /A3 & A4 & A5 & A6 & A7 & A8 & GA2 & /A12I & /S4  
+ /A86 & A87 & /A88 & /A89;

A87 wird gesetzt wenn die Logik

"A0 & A1 & A2 & /A3 & A4 & A5 & A6 & A7 & A8 & GA2 & /A12I & /S4" entspricht "\$8FF7" zutrifft. Bit Darstellung low von links!

1	1	1	0	1	1	1	1	1	1	0	0
Bit0	1	2	3	4	5	6	7	8	9-11	12	\$8000

Die Adresse \$8000 ist aktiv wenn S4=0 ist.

Der zweite Term der Verknüpfung fängt mit ODER "+" an und dient der Selbsthaltung nachdem der erste Term einmal erfüllt war.

Wird eines der anderen Ausgänge A86 - A88 -A89 gesetzt erfolgt automatisch eine Löschung dieser Selbsthaltung.

/A86 & A87 & /A88 & /A89  
0 & 1 & 0 & 0

Im zweiten GAL werden dann die entsprechenden Epromadressen dazu dekodiert.

Nachbau auf Basis einer XEG´s Platine inklusive aller Bauteilen

Benötigte Teile für den Nachbau:

1 \* 64k Eprom vom Typ 27C512, 2\* GAL´s vom Typ 22V10 und 1 altes XEG´s Modul mit geriffelter Oberfläche. Den neuen Schaltplan des Modules und natürlich Erfahrung im Umgang mit dem LötKolben. Die Verbindungen wurden mit lackisolierten Drähten durchgeführt. Hält man den LötKolben eine Zeitlang auf das Drahtende so verbrennt der Lack und das Lot verbindet sich mit dem Draht.

Die Originaldaten vom Rom sind in der Reihenfolge IC1 16K + IC2 16k und IC3 8k in ein File zu kopieren und auf das 27C512 Eprom ohne Offsetwert (\$0) zu brennen. Schneller geht es wenn man das Romfile auf <http://atarimania.com> herunterlädt. Beide GAL´s werden mit den JEDEC Konfigurationsdaten geladen und mit IC2 / IC3 beschriftet damit eine Verwechslung ausgeschlossen ist. Alle Bauteile sind auf der XEG´s Platine zu entfernen, das Rom wird durch das Eprom ersetzt und die Verdrahtung nach dem Schaltplan durchgeführt. Nicht vergessen - die Leiterbahn auf der Platine vom Eprom Pin 2 – A12 zu unterbrechen. Ist nach einer Selbstkontrolle alles richtig angeschlossen dann sollte dem Abschlusstest nichts mehr im Wege stehen.



XEG's Gehäuse mit Platine und allen Bauteilen





Links das Original, rechts der Nachbau

## Das GAL-Listings

\* Version 1.0 Stand 14.04.2007 13:30Uhr  
 Logik Baustein 1 für das Bounty Bob Strikes Back Modul  
 (c) by Bernd Herale  
 Der Nachbau ist ausschließlich nur für den privaten Gebrauch gestattet  
 \*/

CHIP	Gal1_BBSB				GAL22V10 REGISTERED_MODE							
/*1	2	3	4	5	6	7	8	9	10	11	12*/	
A0	A1	A2	A3	A4	A5	A6	A7	A8	S5	GA2	GND	
NC	A86	A96	S4	A12I	A87	A88	A89	A97	A98	A99	VCC	
/*13	14	15	16	17	18	19	20	21	22	23	24 */	

/\* A9, A10, A11 und PHI werden über das 2te GAL geführt - das resultierende Signal kommt als GA2 zurück \*/

/\* Der Grund liegt darin dass in diesem GAL der Speicher für die gesamte Logik nicht ausreicht - Fehlermeldung E1 \*/

/\* Ein Zugriff auf die Adresse \$8FF6 schaltet den Ausgang A86 auf 1 und setzt die anderen A87, A88, A89 zurück \*/

A86 = /A0 & A1 & A2 & /A3 & A4 & A5 & A6 & A7 & A8 & GA2 & /A12I & /S4  
+ A86 & /A87 & /A88 & /A89;

/\* Ein Zugriff auf die Adresse \$8FF7 schaltet den Ausgang A87 auf 1 und setzt die anderen A86, A88, A89 zurück \*/

A87 = A0 & A1 & A2 & /A3 & A4 & A5 & A6 & A7 & A8 & GA2 & /A12I & /S4  
+ /A86 & A87 & /A88 & /A89;

/\* Ein Zugriff auf die Adresse \$8FF8 schaltet den Ausgang A88 auf 1 und setzt die anderen A86, A87, A89 zurück \*/

A88 = /A0 & /A1 & /A2 & A3 & A4 & A5 & A6 & A7 & A8 & GA2 & /A12I & /S4  
+ /A86 & /A87 & A88 & /A89;

/\* Ein Zugriff auf die Adresse \$8FF9 schaltet den Ausgang A89 auf 1 und setzt die anderen A86, A87, A88 zurück \*/

A89 = A0 & /A1 & /A2 & A3 & A4 & A5 & A6 & A7 & A8 & GA2 & /A12I & /S4  
+ /A86 & /A87 & /A88 & A89;

/\* Ein Zugriff auf die Adresse \$9FF6 schaltet den Ausgang A96 auf 1 und setzt die anderen A97, A98, A99 zurück \*/

A96 = /A0 & A1 & A2 & /A3 & A4 & A5 & A6 & A7 & A8 & GA2 & A12I & /S4  
+ A96 & /A97 & /A98 & /A99;

/\* Ein Zugriff auf die Adresse \$9FF7 schaltet den Ausgang A97 auf 1 und setzt die anderen A96, A98, A99 zurück \*/

A97 = A0 & A1 & A2 & /A3 & A4 & A5 & A6 & A7 & A8 & GA2 & A12I & /S4  
+ /A96 & A97 & /A98 & /A99;

/\* Ein Zugriff auf die Adresse \$9FF8 schaltet den Ausgang A98 auf 1 und setzt die anderen A96, A97, A99 zurück \*/

A98 = /A0 & /A1 & /A2 & A3 & A4 & A5 & A6 & A7 & A8 & GA2 & A12I & /S4  
+ /A96 & /A97 & A98 & /A99;

/\* Ein Zugriff auf die Adresse \$9FF9 schaltet den Ausgang A99 auf 1 und setzt die anderen A96, A97, A98 zurück \*/

A99 = A0 & /A1 & /A2 & A3 & A4 & A5 & A6 & A7 & A8 & GA2 & A12I & /S4  
+ /A96 & /A97 & /A98 & A99;

\*

Version 1.0 Stand 14.04.2007 13:30Uhr  
 Logik Baustein 2 für das Bounty Bob Strikes Back Modul  
 (c) by Bernd Herale  
 Der Nachbau ist ausschließlich nur für den privaten Gebrauch gestattet

\*/

CHIP	Gal2_BBSB	GAL22V10 REGISTERED_MODE									
/*1	2	3	4	5	6	7	8	9	10	11	12*/
S4	A99	A98	A97	A89	A88	A87	A12I	A11	A10	A9	GND
S5	A12	A13	A14	A15	CE	NC	NC	PHI	NC	GA2	VCC
/*13	14	15	16	17	18	19	20	21	22	23	24 */

/\* Der Bereich des Eprom ist aufgeteilt in \$0 - \$3FFF 4 Seiten im Speicherbereich \$8000 bis \$8FFF

\$4000-\$7FFF 4 Seiten im Speicherbereich \$9000 bis \$9FFF  
 \$8000-\$9FFF im Speicherbereich \$A000 bis \$BFFF\*/

/\* Verknüpfung mit der Grundlogik von GAL1 da der interne Speicher der Logik nicht ausreicht Fehler E0 \*/

GA2 = A9 & A10 & A11 & PHI;

/\* Gedankenstütze: Der Breich \$8000-8FFF ist aktiv bei A12I=0 und A14=0 - 4 Seiten können umgeschaltet werden

A86 ist A12=0 und A13=0 Seite 1

A87 ist A12=1 und A13=0 Seite 2

A88 ist A12=0 und A13=1 Seite 3

A89 ist A12=1 und A13=1 Seite 4

Bereich \$9000-9FFF ist aktiv bei A12I=1 und A14=1 - 4 Seiten können umgeschaltet werden

A96 ist A12=0 und A13=0 Seite 1

A97 ist A12=1 und A13=0 Seite 2

A98 ist A12=0 und A13=1 Seite 3

A99 ist A12=1 und A13=1 Seite 4

Im Einschaltzustand sind die Ausgangssignale bei A86 und A96 : A12=0 und A13=0.

\*/

A12 = A12I & S4 & /S5 /\* A12I wird auf A12 gesetzt wenn der Speicherbereich \$A000-\$BFFF aktiv ist. \*/

+ /S4 & S5 & A87 & /A12I /\* A12 wird gesetzt wenn A87 = 1 und A12I = 0 (\$8000-\$8FFF) aktiv ist \*/

$+ /S4 \& S5 \& A89 \& /A12I$     /\* A12 wird gesetzt wenn  $A89 = 1$  und  $A12I = 0$   
 (\$8000-\$8FFF) aktiv ist \*/  
 $+ /S4 \& S5 \& A97 \& A12I$     /\* A12 wird gesetzt wenn  $A97 = 1$  und  $A12I = 1$   
 (\$9000-\$9FFF) aktiv ist \*/  
 $+ /S4 \& S5 \& A99 \& A12I;$     /\* A12 wird gesetzt wenn  $A99 = 1$  und  $A12I = 1$   
 (\$9000-\$9FFF) aktiv ist \*/

$A13 = /S4 \& S5 \& A88 \& /A12I$   
 $+ /S4 \& S5 \& A89 \& /A12I$   
 $+ /S4 \& S5 \& A98 \& A12I$   
 $+ /S4 \& S5 \& A99 \& A12I;$

$A14 = /S4 \& S5 \& A12I;$

$A15 = S4 \& /S5;$     /\*  $A15 = 1$  bei Anwahl des Speicherbereiches zwischen  
 \$A000-\$BFFF \*/

$/CE = /S4 + /S5;$     /\* Das Eprom ist aktiv im Bereich \$8000-\$BFFF \*/

Viel Erfolg beim Nachbau,  
 Bernd Herale