bonvech / **SIMP_Megaloader**

Code Issues Pull requests Actions Projects Wiki Security Insights

Add files via upload

Browse files

main

bonvech committed 3 minutes ago

1 parent 7ce0f72 commit 8c8bb00fdd97b104a6703b06b928c70a07f29e19

Showing **1 changed file** with **93 additions** and **95 deletions**.

Unified Split

188 Registers_SIPMZynq.cs

```
20  20          private void RegistersInit_SIPMZynq()
21  21          {
22  22              int num;
    23  +           uint ch_num = 32;
    24  +           uint adc_reg;
    25  +           uint adc_offset = 0x80000;
23  26
24      -               ////***************************** ADC Config **************
    27  +           //*********** Пороги **********//
    28  +           uint thrL = 2148;
    29  +           uint thrH = 2158;
    30  +
    31  +           ////***************************** ADC Config ***********************
25  32                  num = XilWriteReg(0x200018, 0x2810); // пишем во все АЦП о
26  33              num = XilWriteReg(0x20001A, 0x8100); // Bit-wise
27  34              Thread.Sleep(10);
28  35              num = XilWriteReg(0x200018, 0x4610); // пишем во все АЦП одновременн
29  36              num = XilWriteReg(0x20001A, 0x8201); // Enable Reg, 12 bit, two-wire
30  37                  Thread.Sleep(10);
31  38              num = XilWriteReg(0x200018, 0x4210); // пишем во все АЦП одновременн
32      -           num = XilWriteReg(0x20001A, 0x8000); // LCLK Phase
    39  +           num = XilWriteReg(0x20001A, 0x8060); // LCLK Phase
33  40              Thread.Sleep(10);
    41  +           //num = XilWriteReg(0x200018, 0xBE10); // пишем во все АЦП одновреме
    42  +           //num = XilWriteReg(0x20001A, 0x80E7); // LVDS Delay
    43  +           //Thread.Sleep(10);
34  44
35      -           num = XilWriteReg(0x200028, 0x1);   // 50 MHz (0 - 100MHz, 1 - 50 MH
    45  +           num = XilWriteReg(0x200028, 0x0);   // 50 MHz (0 - 100MHz, 1 - 50 MH
36  46                  ////*****************************************************
```

```
37  47
38      -
39      -               num = XilWriteReg(0x90028, 0); // Сглаживание Channel 0
40      -               num = XilWriteReg(0x98028, 0); // Сглаживание Channel 2
41      -
42      -               num = XilWriteReg(0xb0028, 0); // Сглаживание Channel 4
43      -               num = XilWriteReg(0xb8028, 0); // Сглаживание Channel 6
44      -
45      -               num = XilWriteReg(0xd0028, 0); // Сглаживание Channel 8
46      -               num = XilWriteReg(0xd8028, 0); // Сглаживание Channel A
47      -
48      -               //*********** Смещение данных ***********//
49      -               num = XilWriteReg(0x90040, 0); // Data Offset Channel 0
50      -               num = XilWriteReg(0x90042, 0); // Data Offset Channel 1
51      -               num = XilWriteReg(0x98040, 0); // Data Offset Channel 2
52      -               num = XilWriteReg(0x98042, 0); // Data Offset Channel 3
53      -
54      -               num = XilWriteReg(0xb0040, 0); // Data Offset Channel 4
55      -               num = XilWriteReg(0xb0042, 0); // Data Offset Channel 5
56      -               num = XilWriteReg(0xb8040, 0); // Data Offset Channel 6
57      -               num = XilWriteReg(0xb8042, 0); // Data Offset Channel 7
58      -
59      -               num = XilWriteReg(0xd0040, 0); // Data Offset Channel 8
60      -               num = XilWriteReg(0xd0042, 0); // Data Offset Channel 9
61      -               num = XilWriteReg(0xd8040, 0); // Data Offset Channel A
62      -               num = XilWriteReg(0xd8042, 0); // Data Offset Channel B
63      -
64      -               //*********** Длина данных ***********//
65      -               //num = XilWriteReg(0x90044, 4); // Data Length Channel 0
66      -               //num = XilWriteReg(0x90046, 4); // Data Length Channel 1
67      -               //num = XilWriteReg(0x98044, 4); // Data Length Channel 2
68      -               //num = XilWriteReg(0x98046, 4); // Data Length Channel 3
69      -
70      -               //num = XilWriteReg(0xb0044, 4); // Data Length Channel 4
71      -               //num = XilWriteReg(0xb0046, 4); // Data Length Channel 5
72      -               //num = XilWriteReg(0xb8044, 4); // Data Length Channel 6
73      -               //num = XilWriteReg(0xb8046, 4); // Data Length Channel 7
74      -
75      -               //num = XilWriteReg(0xd0044, 4); // Data Length Channel 8
76      -               //num = XilWriteReg(0xd0046, 4); // Data Length Channel 9
77      -               //num = XilWriteReg(0xd8044, 4); // Data Length Channel A
78      -               //num = XilWriteReg(0xd8046, 4); // Data Length Channel B
79      -
80      -               num = XilWriteReg(0x90044, 2048); // Data Length Channel 0
81      -               num = XilWriteReg(0x90046, 2048); // Data Length Channel 1
82      -               num = XilWriteReg(0x98044, 2048); // Data Length Channel 2
83      -               num = XilWriteReg(0x98046, 2048); // Data Length Channel 3
```

```
-
-            num = XilWriteReg(0xb0044, 2048); // Data Length Channel 4
-            num = XilWriteReg(0xb0046, 2048); // Data Length Channel 5
-            num = XilWriteReg(0xb8044, 2048); // Data Length Channel 6
-            num = XilWriteReg(0xb8046, 2048); // Data Length Channel 7
-
-            num = XilWriteReg(0xd0044, 2048); // Data Length Channel 8
-            num = XilWriteReg(0xd0046, 2048); // Data Length Channel 9
-            num = XilWriteReg(0xd8044, 2048); // Data Length Channel A
-            num = XilWriteReg(0xd8046, 2048); // Data Length Channel B
+            for (uint i = 0; i < ch_num; i++)
+            {
+                adc_reg = (i << 12) | adc_offset;
+                if (i <12) num = XilWriteReg(adc_reg + 0x0, 0); // Request Disab
+                else num = XilWriteReg(adc_reg + 0x0, 4); // Request Disable = 1
+                num = XilWriteReg(adc_reg + 0x28, 0); // Сглаживание
+                num = XilWriteReg(adc_reg + 0x40, 8); // Data Offset
+                num = XilWriteReg(adc_reg + 0x44, 2048); // Data Length
+                num = XilWriteReg(adc_reg + 0xE, 512); // Stop Delay
+                num = XilWriteReg(adc_reg + 0x10, thrL); // Threshold L
+                num = XilWriteReg(adc_reg + 0x20, thrH); // Threshold H
+            }
+
+            ////************ Смещение данных ***********//
+            //num = XilWriteReg(0x90040, 0); // Data Offset Channel 0
+            //num = XilWriteReg(0x90042, 0); // Data Offset Channel 1
+            //num = XilWriteReg(0x98040, 0); // Data Offset Channel 2
+            //num = XilWriteReg(0x98042, 0); // Data Offset Channel 3
+
+            //num = XilWriteReg(0xb0040, 0); // Data Offset Channel 4
+            //num = XilWriteReg(0xb0042, 0); // Data Offset Channel 5
+            //num = XilWriteReg(0xb8040, 0); // Data Offset Channel 6
+            //num = XilWriteReg(0xb8042, 0); // Data Offset Channel 7
+
+            //num = XilWriteReg(0xd0040, 0); // Data Offset Channel 8
+            //num = XilWriteReg(0xd0042, 0); // Data Offset Channel 9
+            //num = XilWriteReg(0xd8040, 0); // Data Offset Channel A
+            //num = XilWriteReg(0xd8042, 0); // Data Offset Channel B
+
+            ////************ Длина данных ***********//
+            //num = XilWriteReg(0x90044, 2048); // Data Length Channel 0
+            //num = XilWriteReg(0x90046, 2048); // Data Length Channel 1
+            //num = XilWriteReg(0x98044, 2048); // Data Length Channel 2
+            //num = XilWriteReg(0x98046, 2048); // Data Length Channel 3
+
+            //num = XilWriteReg(0xb0044, 2048); // Data Length Channel 4
+            //num = XilWriteReg(0xb0046, 2048); // Data Length Channel 5
```

```
+            //num = XilWriteReg(0xb8044, 2048); // Data Length Channel 6
+            //num = XilWriteReg(0xb8046, 2048); // Data Length Channel 7
+
+            //num = XilWriteReg(0xd0044, 2048); // Data Length Channel 8
+            //num = XilWriteReg(0xd0046, 2048); // Data Length Channel 9
+            //num = XilWriteReg(0xd8044, 2048); // Data Length Channel A
+            //num = XilWriteReg(0xd8046, 2048); // Data Length Channel B


                    //num = XilWriteReg(0x9000e, 100); // Stop Delay
                    //num = XilWriteReg(0x9800e, 100); // Stop Delay
                    //num = XilWriteReg(0xd000e, 0); // Stop Delay
                    //num = XilWriteReg(0xd800e, 0); // Stop Delay

-                   //*********** Пороги ***********//
-            uint thrL = 2148;
-            uint thrH = 2158;
-
-                    num = XilWriteReg(0x90010, thrL); // Threshold 0 L
-                    num = XilWriteReg(0x90014, thrL); // Threshold 1 L
-                    num = XilWriteReg(0x90020, thrH); // Threshold 0 H
-                    num = XilWriteReg(0x90024, thrH); // Threshold 1 H
-                    num = XilWriteReg(0x98010, thrL); // Threshold 2 L
-                    num = XilWriteReg(0x98014, thrL); // Threshold 3 L
-                    num = XilWriteReg(0x98020, thrH); // Threshold 2 H
-                    num = XilWriteReg(0x98024, thrH); // Threshold 3 H
-
-                    num = XilWriteReg(0xb0010, thrL); // Threshold 4 L
-                    num = XilWriteReg(0xb0014, thrL); // Threshold 5 L
-                    num = XilWriteReg(0xb0020, thrH); // Threshold 4 H
-                    num = XilWriteReg(0xb0024, thrH); // Threshold 5 H
-                    num = XilWriteReg(0xb8010, thrL); // Threshold 6 L
-                    num = XilWriteReg(0xb8014, thrL); // Threshold 7 L
-                    num = XilWriteReg(0xb8020, thrH); // Threshold 6 H
-                    num = XilWriteReg(0xb8024, thrH); // Threshold 7 H
-
-                    num = XilWriteReg(0xd0010, thrL); // Threshold 8 L
-                    num = XilWriteReg(0xd0014, thrL); // Threshold 9 L
-                    num = XilWriteReg(0xd0020, thrH); // Threshold 8 H
-                    num = XilWriteReg(0xd0024, thrH); // Threshold 9 H
-                    num = XilWriteReg(0xd8010, thrL); // Threshold A L
-                    num = XilWriteReg(0xd8014, thrL); // Threshold B L
-                    num = XilWriteReg(0xd8020, thrH); // Threshold A H
-                    num = XilWriteReg(0xd8024, thrH); // Threshold B H
-            //***********************//
-
-                    //num = XilWriteReg(0x20000C, 0xffff); //маска каналов
-                    //num = XilWriteReg(0x20000D, 0xffff); //маска каналов
```

```
143        -
144        -                    num = XilWriteReg(0x20000C, 0x80); //маска каналов
107    +           ////*********** Пороги ***********//
108    +           //uint thrL = 2148;
109    +           //uint thrH = 2158;
110    +
111    +           //num = XilWriteReg(0x90010, thrL); // Threshold 0 L
112    +           //num = XilWriteReg(0x90014, thrL); // Threshold 1 L
113    +           //num = XilWriteReg(0x90020, thrH); // Threshold 0 H
114    +           //num = XilWriteReg(0x90024, thrH); // Threshold 1 H
115    +           //num = XilWriteReg(0x98010, thrL); // Threshold 2 L
116    +           //num = XilWriteReg(0x98014, thrL); // Threshold 3 L
117    +           //num = XilWriteReg(0x98020, thrH); // Threshold 2 H
118    +           //num = XilWriteReg(0x98024, thrH); // Threshold 3 H
119    +
120    +           //num = XilWriteReg(0xb0010, thrL); // Threshold 4 L
121    +           //num = XilWriteReg(0xb0014, thrL); // Threshold 5 L
122    +           //num = XilWriteReg(0xb0020, thrH); // Threshold 4 H
123    +           //num = XilWriteReg(0xb0024, thrH); // Threshold 5 H
124    +           //num = XilWriteReg(0xb8010, thrL); // Threshold 6 L
125    +           //num = XilWriteReg(0xb8014, thrL); // Threshold 7 L
126    +           //num = XilWriteReg(0xb8020, thrH); // Threshold 6 H
127    +           //num = XilWriteReg(0xb8024, thrH); // Threshold 7 H
128    +
129    +           //num = XilWriteReg(0xd0010, thrL); // Threshold 8 L
130    +           //num = XilWriteReg(0xd0014, thrL); // Threshold 9 L
131    +           //num = XilWriteReg(0xd0020, thrH); // Threshold 8 H
132    +           //num = XilWriteReg(0xd0024, thrH); // Threshold 9 H
133    +           //num = XilWriteReg(0xd8010, thrL); // Threshold A L
134    +           //num = XilWriteReg(0xd8014, thrL); // Threshold B L
135    +           //num = XilWriteReg(0xd8020, thrH); // Threshold A H
136    +           //num = XilWriteReg(0xd8024, thrH); // Threshold B H
137    +           ////***********************//
138    +
139    +                    num = XilWriteReg(0x20000C, 0xfff); //маска каналов
145    140                  num = XilWriteReg(0x20000D, 0x0); //маска каналов
146    141
142    +           //num = XilWriteReg(0x20000C, 0x80); //маска каналов
143    +           //num = XilWriteReg(0x20000D, 0x0); //маска каналов
144    +
147    145          //num = XilWriteReg(0x200006, 0x101); // кратность совпадений и внеш
148    146          //                      num = XilWriteReg(0x200006, 0x301); // кра
149    147                   num = XilWriteReg(0x200006, 0x1); // кратность совпадений
```

0 comments on commit `8c8bb00`