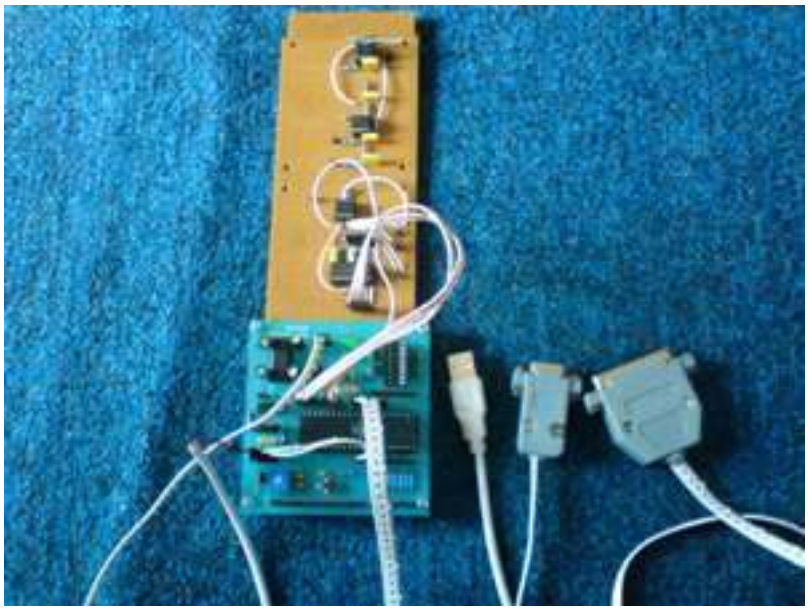


Foto Rangkaian Modulator Digital

Data ROM Sinus

No	Data (Hex)	No	Data (Hex)	No	Data (hex)	No	Data (Hex)
1	80	36	E0	71	FE	106	C4
2	83	37	E2	72	FE	107	C1
3	86	38	E4	73	FD	108	BF
4	89	39	E6	74	FC	109	BC
5	8C	40	E8	75	FC	110	B9
6	8F	41	EA	76	FB	111	B6
7	92	42	EC	77	FA	112	B3
8	95	43	ED	78	F9	113	B0
9	98	44	EF	79	F8	114	AE
10	9C	45	F0	80	F7	115	AB
11	9F	46	F2	81	F6	116	A8
12	A2	47	F3	82	F5	117	A5
13	A5	48	F5	83	F3	118	A2
14	A8	49	F6	84	F2	119	9F
15	AB	50	F7	85	F0	120	9C
16	AE	51	F8	86	EF	121	98
17	B0	52	F9	87	ED	122	95
18	B3	53	FA	88	EC	123	92
19	B6	54	FB	89	EA	124	8F
20	B9	55	FC	90	E8	125	8C
21	BC	56	FC	91	E6	126	89
22	BF	57	FD	92	E4	127	86
23	C1	58	FE	93	E2	128	83
24	C4	59	FE	94	E0	129	80
25	C7	60	FF	95	DE	130	80
26	C9	61	FF	96	DC	131	7C
27	CC	62	FF	97	DA	132	79
28	CE	63	FF	98	D8	133	73
29	D1	64	FF	99	D5	134	70
30	D3	65	FF	100	D3	135	6D
31	D5	66	FF	101	D1	136	6A
32	D8	67	FF	102	CE	137	67
33	DA	68	FF	103	CC	138	63
34	DC	69	FF	104	C9	139	60
35	DE	70	FF	105	C7	140	5D

Data ROM Sinus

No	Data (Hex)	No	Data (Hex)	No	Data (Hex)	No	Data (Hex)
141	5A	176	0A	211	0C	246	5D
142	57	177	09	212	0D	247	60
143	54	178	08	213	0F	248	63
144	51	179	07	214	10	249	67
145	4F	180	06	215	12	250	6A
146	4C	181	05	216	13	251	6D
147	49	182	04	217	15	252	70
148	46	183	03	218	17	253	73
149	43	184	03	219	19	254	76
150	40	185	02	220	1B	255	79
151	3E	186	01	221	1D	256	7C
152	3B	187	00	222	1F		
153	38	188	00	223	21		
154	36	189	00	224	23		
155	33	190	00	225	25		
156	31	191	00	226	27		
157	2E	192	00	227	2A		
158	2C	193	00	228	2C		
159	2A	194	00	229	2E		
160	27	195	00	230	31		
161	25	196	00	231	33		
162	23	197	00	232	36		
163	21	198	00	233	38		
164	1F	199	01	234	3B		
165	1F	200	01	235	3E		
166	1D	201	02	236	40		
167	19	202	03	237	43		
168	17	203	03	238	46		
169	15	204	04	239	49		
170	13	205	05	240	4C		
171	12	206	06	241	4F		
172	10	207	07	242	51		
173	0F	208	08	243	54		
174	0D	209	09	244	57		
175	0C	210	0A	245	5A		

T

a

bel Keluaran RO Sinus untuk Input 00

No	Data (Hex)	No	Data (Hex)	No	Data (hex)	No	Data (Hex)
1	25	36	f0	71	2	106	f0
2	1d	37	f6	72	0	107	f6
3	15	38	fa	73	0	108	fa
4	f	39	fd	74	0	109	fd
5	9	40	ff	75	2	110	ff
6	5	41	ff	76	5	111	ff
7	2	42	ff	77	9	112	25
8	0	43	fd	78	f	113	1d
9	0	44	fa	79	15	114	15
10	0	45	f6	80	1f	115	f
11	2	46	f0	81	25	116	9
12	5	47	ea	82	2e	117	5
13	9	48	e2	83	38	118	2
14	f	49	da	84	43	119	0
15	15	50	d1	85	4f	120	0
16	1f	51	c7	86	5a	121	0
17	25	52	bc	87	67	122	2
18	2e	53	b0	88	73	123	5
19	38	54	a5	89	80	124	9
20	43	55	98	90	8c	125	f
21	4f	56	8c	91	98	126	15
22	5a	57	80	92	a5	127	1f
23	67	58	73	93	b0	128	25
24	73	59	67	94	bc	129	2e
25	80	60	5a	95	c7	130	38
26	8c	61	4f	96	d1	131	43
27	98	62	43	97	da	132	4f
28	a5	63	38	98	e2	133	5a
29	b0	64	2e	99	ea	134	67
30	bc	65	25	100	f0	135	73
31	c7	66	1d	101	f6	136	80
32	d1	67	15	102	fa	137	8c
33	da	68	f	103	fd	138	98
34	e2	69	9	104	ff	139	a5
35	ea	70	5	105	ff	140	b0

T
a

bel Keluaran ROM Sinus untuk Input 00

No	Data (Hex)	No	Data (Hex)	No	Data (Hex)	No	Data (Hex)
141	bc	176	25	211	f0	246	73
142	c7	177	1d	212	f6	247	80
143	d1	178	15	213	fa	248	8c
144	da	179	f	214	fd	249	98
145	e2	180	9	215	ff	250	a5
146	ea	181	5	216	ff	251	b0
147	f0	182	2	217	f0	252	bc
148	f6	183	0	218	f6	253	c7
149	fa	184	0	219	fa	254	d1
150	fd	185	0	220	fd	255	da
151	ff	186	2	221	ff	256	e2
152	ff	187	5	222	ff		
153	ff	188	9	223	25		
154	fd	189	f	224	1d		
155	fa	190	15	225	15		
156	f6	191	1f	226	f		
157	f0	192	25	227	9		
158	ea	193	2e	228	5		
159	e2	194	38	229	2		
160	da	195	43	230	0		
161	d1	196	4f	231	0		
162	c7	197	5a	232	0		
163	bc	198	67	233	2		
164	b0	199	73	234	5		
165	a5	200	80	235	9		
166	98	201	8c	236	f		
167	8c	202	98	237	15		
168	80	203	a5	238	1f		
169	73	204	b0	239	25		
170	67	205	bc	240	2e		
171	5a	206	c7	241	38		
172	4f	207	d1	242	43		
173	43	208	da	243	4f		
174	38	209	e2	244	5a		
175	2e	210	ea	245	67		

T

a

bel Keluaran ROM Sinus untuk Input 01

No	Data (Hex)	No	Data (Hex)	No	Data (hex)	No	Data (Hex)
1	25	36	bc	71	67	106	73
2	2e	37	b0	72	73	107	67
3	38	38	a5	73	80	108	5a
4	43	39	98	74	8c	109	4f
5	4f	40	8c	75	98	110	43
6	5a	41	80	76	a5	111	38
7	67	42	73	77	b0	112	2e
8	73	43	67	78	bc	113	25
9	80	44	5a	79	c7	114	1d
10	8c	45	4f	80	d1	115	15
11	98	46	43	81	da	116	f
12	a5	47	38	82	e2	117	9
13	b0	48	2e	83	ea	118	5
14	bc	49	25	84	f0	119	2
15	c7	50	1d	85	f6	120	0
16	d1	51	15	86	fa	121	0
17	da	52	f	87	fd	122	0
18	e2	53	9	88	ff	123	2
19	ea	54	5	89	ff	124	5
20	f0	55	2	90	ff	125	9
21	f6	56	0	91	fd	126	f
22	fa	57	0	92	fa	127	15
23	fd	58	0	93	f6	128	1f
24	ff	59	2	94	f0	129	25
25	ff	60	5	95	ea	130	2e
26	ff	61	9	96	e2	131	38
27	fd	62	f	97	da	132	43
28	fa	63	15	98	d1	133	4f
29	f6	64	1f	99	c7	134	5a
30	f0	65	25	100	bc	135	67
31	ea	66	2e	101	b0	136	73
32	e2	67	38	102	a5	137	80
33	da	68	43	103	98	138	8c
34	d1	69	4f	104	8c	139	98
35	c7	70	5a	105	80	140	a5

T

a

bel Keluaran ROM Sinus untuk Input 01

No	Data (Hex)	No	Data (Hex)	No	Data (Hex)	No	Data (Hex)
141	b0	176	2e	211	ea	246	5
142	bc	177	25	212	f0	247	2
143	c7	178	1d	213	f6	248	0
144	d1	179	15	214	fa	249	0
145	da	180	f	215	fd	250	0
146	e2	181	9	216	ff	251	2
147	ea	182	5	217	ff	252	5
148	f0	183	2	218	ff	253	9
149	f6	184	0	219	fd	254	f
150	fa	185	0	220	fa	255	15
151	fd	186	0	221	f6	256	1f
152	ff	187	2	222	f0		
153	ff	188	5	223	ea		
154	ff	189	9	224	e2		
155	fd	190	f	225	da		
156	fa	191	15	226	d1		
157	f6	192	1f	227	c7		
158	f0	193	25	228	bc		
159	ea	194	2e	229	b0		
160	e2	195	38	230	a5		
161	da	196	43	231	98		
162	d1	197	4f	232	8c		
163	c7	198	5a	233	80		
164	bc	199	67	234	73		
165	b0	200	73	235	67		
166	a5	201	80	236	5a		
167	98	202	8c	237	4f		
168	8c	203	98	238	43		
169	80	204	a5	239	38		
170	73	205	b0	240	2e		
171	67	206	bc	241	25		
172	5a	207	c7	242	1d		
173	4f	208	d1	243	15		
174	43	209	da	244	f		
175	38	210	e2	245	9		

T

a

bel Keluaran ROM Sinus untuk Input 10

No	Data (Hex)	No	Data (Hex)	No	Data (hex)	No	Data (Hex)
1	da	36	43	71	98	106	8c
2	d1	37	4f	72	8c	107	98
3	c7	38	5a	73	80	108	a5
4	bc	39	67	74	73	109	b0
5	b0	40	73	75	67	110	bc
6	a5	41	80	76	5a	111	c7
7	98	42	8c	77	4f	112	d1
8	8c	43	98	78	43	113	da
9	80	44	a5	79	38	114	e2
10	73	45	b0	80	2e	115	ea
11	67	46	bc	81	25	116	f0
12	5a	47	c7	82	1f	117	f6
13	4f	48	d1	83	15	118	fa
14	43	49	da	84	f	119	fd
15	38	50	e2	85	9	120	ff
16	2e	51	ea	86	5	121	ff
17	25	52	f0	87	2	122	ff
18	1f	53	f6	88	0	123	fd
19	15	54	fa	89	0	124	fa
20	f	55	fd	90	0	125	f6
21	9	56	ff	91	2	126	f0
22	5	57	ff	92	5	127	ea
23	2	58	ff	93	9	128	e2
24	0	59	fd	94	f	129	da
25	0	60	fa	95	15	130	d1
26	0	61	f6	96	1d	131	c7
27	2	62	f0	97	25	132	bc
28	5	63	ea	98	2e	133	b0
29	9	64	e2	99	38	134	a5
30	f	65	da	100	43	135	98
31	15	66	d1	101	4f	136	8c
32	1d	67	c7	102	5a	137	80
33	25	68	bc	103	67	138	73
34	2e	69	b0	104	73	139	67
35	38	70	a5	105	80	140	5a

T

a

bel Keluaran ROM Sinus untuk Input 10

No	Data (Hex)	No	Data (Hex)	No	Data (Hex)	No	Data (Hex)
141	4f	176	d1	211	15	246	fa
142	43	177	da	212	f	247	fd
143	38	178	e2	213	9	248	ff
144	2e	179	ea	214	5	249	ff
145	25	180	f0	215	2	250	ff
146	1f	181	f6	216	0	251	fd
147	15	182	fa	217	0	252	fa
148	f	183	fd	218	0	253	f6
149	9	184	ff	219	2	254	f0
150	5	185	ff	220	5	255	ea
151	2	186	ff	221	9	256	e2
152	0	187	fd	222	f		
153	0	188	fa	223	15		
154	0	189	f6	224	1d		
155	2	190	f0	225	25		
156	5	191	ea	226	2e		
157	9	192	e2	227	38		
158	f	193	da	228	43		
159	15	194	d1	229	4f		
160	1d	195	c7	230	5a		
161	25	196	bc	231	67		
162	2e	197	b0	232	73		
163	38	198	a5	233	80		
164	43	199	98	234	8c		
165	4f	200	8c	235	98		
166	5a	201	80	236	a5		
167	67	202	73	237	b0		
168	73	203	67	238	bc		
169	80	204	5a	239	c7		
170	8c	205	4f	240	d1		
171	98	206	43	241	da		
172	a5	207	38	242	e2		
173	b0	208	2e	243	ea		
174	bc	209	25	244	f0		
75	c7	210	1f	245	f6		

T

a

bel Keluaran ROM Sinus untuk Input 11

No	Data (Hex)	No	Data (Hex)	No	Data (hex)	No	Data (Hex)
1	da	36	f	71	fd	106	0
2	e2	37	9	72	ff	107	2
3	ea	38	5	73	ff	108	5
4	f0	39	2	74	ff	109	9
5	f6	40	0	75	fd	110	f
6	fa	41	0	76	fa	111	15
7	fd	42	0	77	f6	112	1d
8	ff	43	2	78	f0	113	25
9	ff	44	5	79	ea	114	2e
10	ff	45	9	80	e2	115	38
11	fd	46	f	81	da	116	43
12	fa	47	15	82	d1	117	4f
13	f6	48	1d	83	c7	118	5a
14	f0	49	25	84	bc	119	67
15	ea	50	2e	85	b0	120	73
16	e2	51	38	86	a5	121	80
17	da	52	43	87	98	122	8c
18	d1	53	4f	88	8c	123	98
19	c7	54	5a	89	80	124	a5
20	bc	55	67	90	73	125	b0
21	b0	56	73	91	67	126	bc
22	a5	57	80	92	5a	127	c7
23	98	58	8c	93	4f	128	d1
24	8c	59	98	94	43	129	da
25	80	60	a5	95	38	130	e2
26	73	61	b0	96	2e	131	ea
27	67	62	bc	97	25	132	f0
28	5a	63	c7	98	1f	133	f6
29	4f	64	d1	99	15	134	fa
30	43	65	da	100	f	135	fd
31	38	66	e2	101	9	136	ff
32	2e	67	ea	102	5	137	ff
33	25	68	f0	103	2	138	ff
34	1f	69	f6	104	0	139	fd
35	15	70	fa	105	0	140	fa

T

a

bel Keluaran ROM Sinus untuk Input 11

No	Data (Hex)	No	Data (Hex)	No	Data (Hex)	No	Data (Hex)
141	f6	176	1d	211	c7	246	5a
142	f0	177	25	212	bc	247	67
143	ea	178	2e	213	b0	248	73
144	e2	179	38	214	a5	249	80
145	da	180	43	215	98	250	8c
146	d1	181	4f	216	8c	251	98
147	c7	182	5a	217	80	252	a5
148	bc	183	67	218	73	253	b0
149	b0	184	73	219	67	254	bc
150	a5	185	80	220	5a	255	c7
151	98	186	8c	221	4f	256	d1
152	8c	187	98	222	43		
153	80	188	a5	223	38		
154	73	189	b0	224	2e		
155	67	190	bc	225	25		
156	5a	191	c7	226	1f		
157	4f	192	d1	227	15		
158	43	193	da	228	f		
159	38	194	e2	229	9		
160	2e	195	ea	230	5		
161	25	196	f0	231	2		
162	1f	197	f6	232	0		
163	15	198	fa	233	0		
164	f	199	fd	234	0		
165	9	200	ff	235	2		
166	5	201	ff	236	5		
167	2	202	ff	237	9		
168	0	203	fd	238	f		
169	0	204	fa	239	15		
170	0	205	f6	240	1d		
171	2	206	f0	241	25		
172	5	207	ea	242	2e		
173	9	208	e2	243	38		
174	f	209	da	244	43		
175	15	210	d1	245	4f		

/******
******

```
*****************************************
This program was produced by the
CodeWizardAVR V1.25.3 Standard
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :Modulator QPSK
Version :
Date    :17/06/2009
Author  :Jhon Presley Sinaga
Company :NATO
Comments:


Chip type           : ATmega8535
Program type        : Application
Clock frequency     : 11,059200 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 128
*****************************************************/

#include <mega8535.h>

#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// Write a character to the USART Transmitter
#ifndef _DEBUG_TERMINAL_IO_
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while ((UCSRA & DATA_REGISTER_EMPTY)==0);
UDR=c;
}
#pragma used-
#endif
                                                                    //
                                                                    Standar
```

```
d Input/Output functions
#include <stdio.h>
#include <stdio.h>
#include <string.h>
#include <delay.h>

// Declare your global variables here
void init_system(void);
void periksa_ROM(void);
void ambil_data_ROM_SINUS(void);
void ambil_data_ROM_COSINUS(void);

eeprom unsigned char ROM_SINUS[256]=
{0x80, 0x83, 0x86, 0x89, 0x8c, 0x8f, 0x92, 0x95, 0x98, 0x9c, 0x9f, 0xa2, 0xa5,
0xa8, 0xab, 0xae, 0xb0, 0xb3, 0xb6, 0xb9, 0xbc,   0xbf, 0xc1, 0xc4, 0xc7, 0xc9,
0xcc, 0xce, 0xd1, 0xd3, 0xd5, 0xd8, 0xda, 0xdc, 0xde, 0xe0, 0xe2, 0xe4, 0xe6,
0xe8, 0xea, 0xec, 0xed, 0xef, 0xf0, 0xf2, 0xf3, 0xf5, 0xf6, 0xf7, 0xf8, 0xf9, 0xfa,
0xfb, 0xfc, 0xfc, 0xfd, 0xfe, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xfe, 0xfe, 0xfd, 0xfc, 0xfc, 0xfb, 0xfa, 0xf9, 0xf8, 0xf7, 0xf6, 0xf5,
0xf3, 0xf2, 0xf0, 0xef, 0xed, 0xec, 0xea, 0xe8, 0xe6, 0xe4, 0xe2, 0xe0, 0xde, 0xdc,
0xda, 0xd8, 0xd5, 0xd3, 0xd1, 0xce, 0xcc, 0xc9, 0xc7, 0xc4, 0xc1, 0xbf, 0xbc,
0xb9, 0xb6, 0xb3, 0xb0, 0xae, 0xab, 0xa8, 0xa5, 0xa2, 0x9f, 0x9c, 0x98, 0x95,
0x92, 0x8f, 0x8c, 0x89, 0x86, 0x83, 0x80, 0x80, 0x7c, 0x79, 0x73, 0x70, 0x6d,
0x6a, 0x67, 0x63, 0x60, 0x5d, 0x5a, 0x57, 0x54, 0x51, 0x4f, 0x4c, 0x49, 0x46,
0x43, 0x40, 0x3e, 0x3b, 0x38, 0x36, 0x33, 0x31, 0x2e, 0x2c, 0x2a, 0x27, 0x25,
0x23, 0x21, 0x1f, 0x1f, 0x1b, 0x19, 0x17, 0x15, 0x13, 0x12, 0x10, 0x0f, 0x0d,
0x0c, 0x0a, 0x09, 0x08, 0x07, 0x06, 0x05, 0x04, 0x03, 0x03, 0x02, 0x01, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01,
0x02, 0x03, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0c, 0x0d, 0x0f,
0x10, 0x12, 0x13, 0x15, 0x17, 0x19, 0x1b, 0x1d, 0x1f, 0x21, 0x23, 0x25, 0x27,
0x2a, 0x2c, 0x2e, 0x31, 0x33, 0x36, 0x38, 0x3b, 0x3e, 0x40, 0x43, 0x46, 0x49,
0x4c, 0x4f, 0x51, 0x54, 0x57, 0x5a, 0x5d, 0x60, 0x63, 0x67, 0x6a, 0x6d, 0x70,
0x73,  0x76, 0x79, 0x7c };


bit bit_input1, bit_input2;
int l, m,n,o;
unsigned char out_sin, out_cos;
unsigned char out;

void main(void)
{
init_system();
```

while
(1)

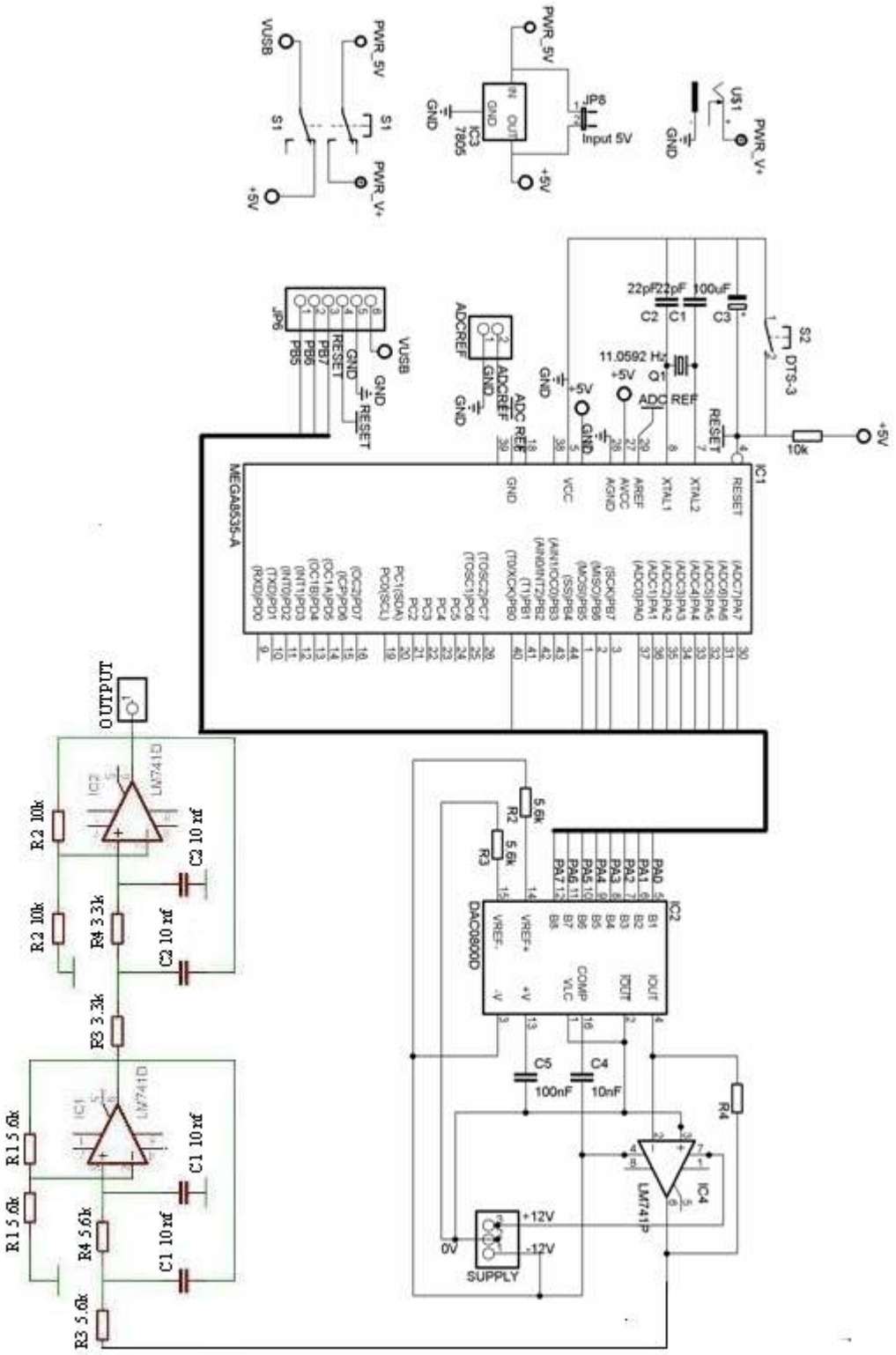
```
    {
    // Place your code here
    bit_input1=PinB.0;
    delay_us(420);
    bit_input2=PinB.0;
    delay_us(420);
    periksa_ROM();
    };
}

void init_system(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTA=0x00;
DDRA=0xFF;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;


// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
                                                                TCCR0
                                                                =0x00;
```

```
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
```

```
initialization
// Communication Parameters: 8 Data, 2 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 19200
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x8E;
UBRRH=0x00;
UBRRL=0x23;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
}

void periksa_ROM(void)
{
int i;
l=28;
m=92;
n=164;
o=228;


for (i=1; i<=64; i++)
    {
    ambil_data_ROM_SINUS();
    //ambil_data_ROM_COSINUS();
    PORTA=out;
    printf("%x \n",out);
    };
}
void ambil_data_ROM_SINUS(void)
{
if (bit_input1==1 && bit_input2==1)
   {
   l=l+4;
   if (l>255)
   {
   l=0;
   }
   out=ROM_SINUS[l];

   };
```

```
                                                                    if
                                                                    (bit_inp
```

```
ut1==0 && bit_input2==1)
   {
   m=m+4;
   if (m>255)
      {
      m=0;
      }
   out=ROM_SINUS[m];
   };
if (bit_input1==1 && bit_input2==0)
{
n=n-4;
   if (n<0)
      {
      n=252;
      }
   out=ROM_SINUS[n];
}
if (bit_input1==0 && bit_input2==0)
{
o=o-4;
   if (o<0)
      {
      o=252;
      }
   out=ROM_SINUS[o];
   }
return;
}
```


VUSB
S1
PWR_5V
PWR_V+
+5V
IC3
7805
IN
OUT
GND
JP6
Input 5V
U$1
S2
DTS-3
10k
22pF
100uF
C1
C2
C3
11.0592 Hz
Q1
ADC REF
ADCREF
RESET
JP5
PB5
PB6
PB7
MEGA8535-A
IC1
OUTPUT
LM741D
R2 10k
R4 3.3k
C2 10 nf
R3 3.3k
R1 5.6k
R4 5.6k
C1 10 nf
R3 5.6k
5.6k
DAC0800D
IC2
C5
100nF
C4
10nF
R4
LM741P
IC4
SUPPLY
0V
+12V
-12V

Featu

res

- High-performance, Low-power AVR®8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 8K Bytes of In-System Self-Programmable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 512 Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 512 Bytes Internal SRAM
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    8 Single-ended Channels
    7 Differential Channels for TQFP Package Only
    2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega8535L
  - 4.5 - 5.5V for ATmega8535
- Speed Grades
  - 0 - 8 MHz for ATmega8535L
  - 0 - 16 MHz for ATmega8535

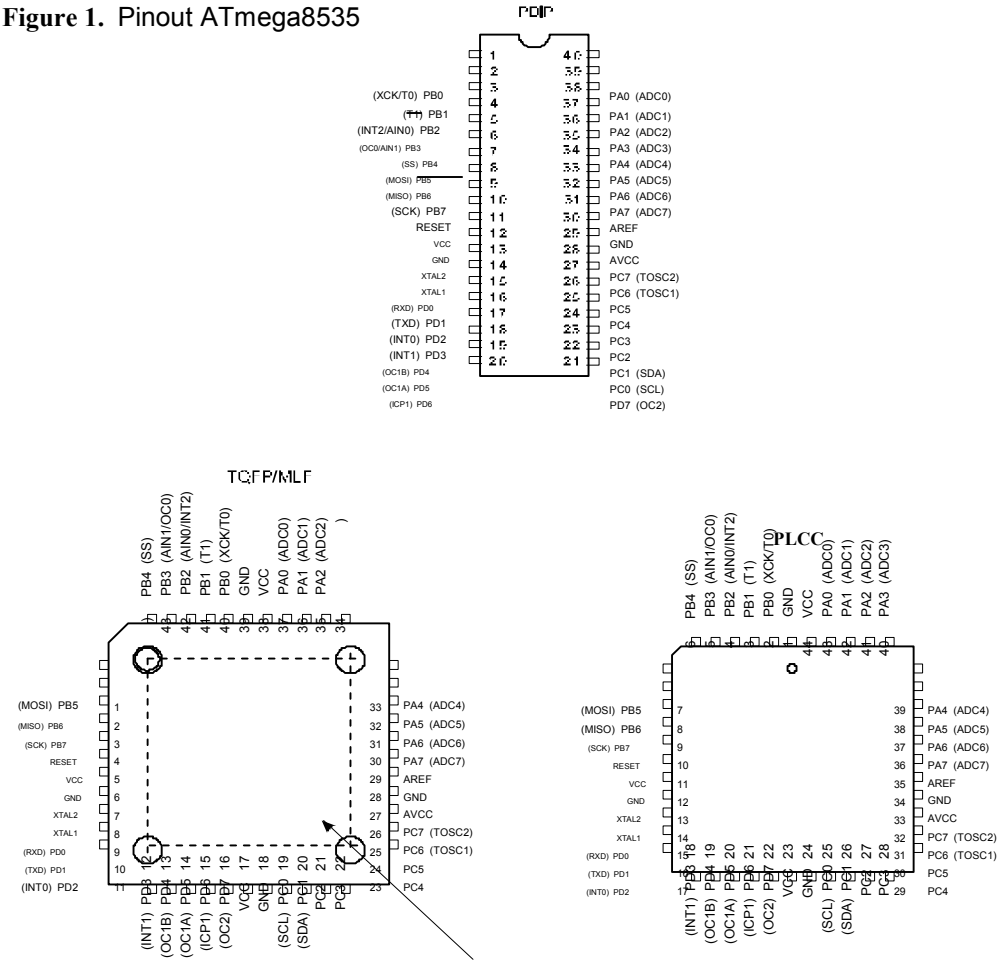
AVR

8-bitMicrocontroller with 8K Bytes In-System Programmable Flash

ATmega8535 ATmega8535L

Pin Configurations

Figure 1. Pinout ATmega8535



NOTE: MLF Bottom pad should be soldered to ground.

Disclaimer

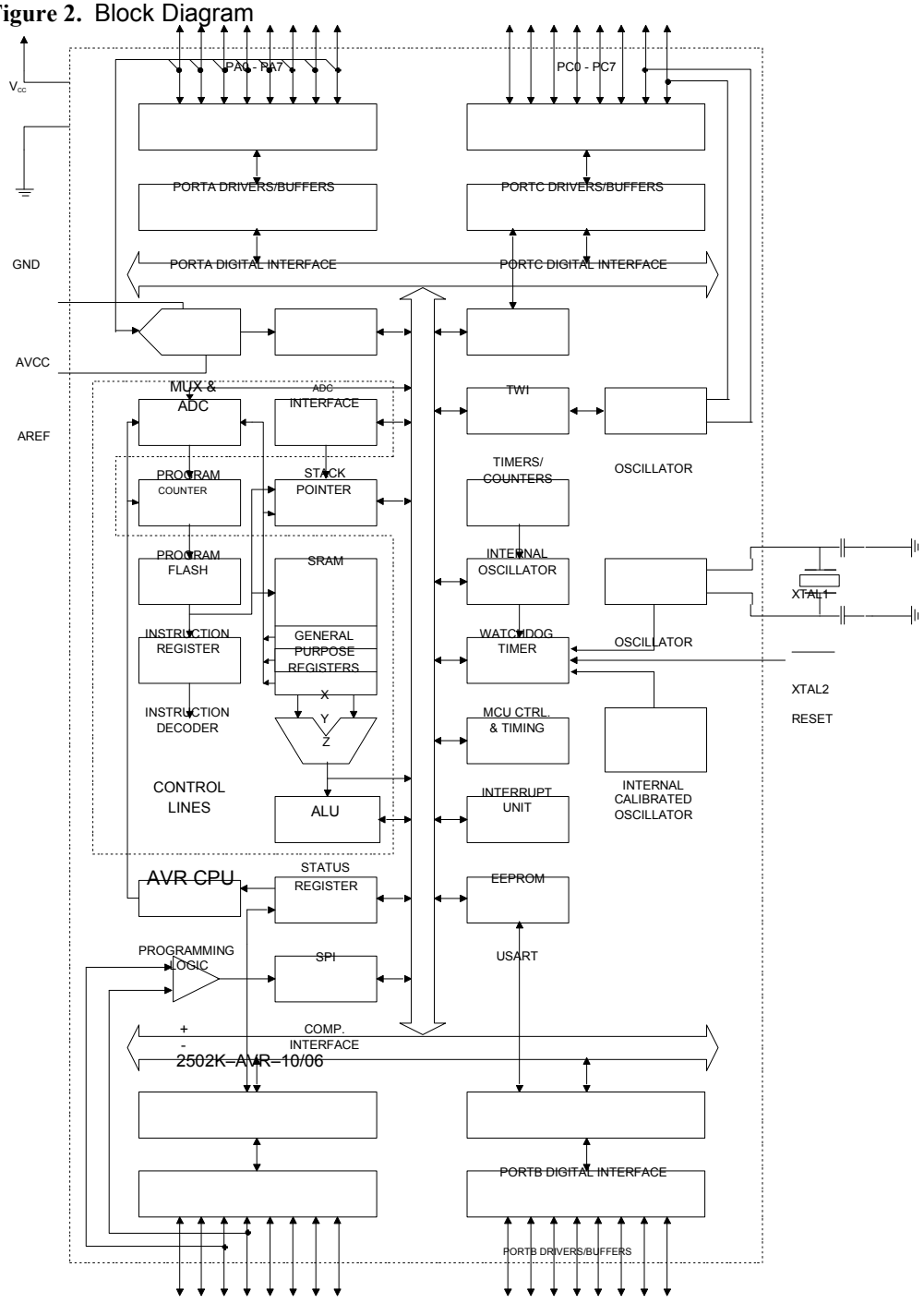
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega8535 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing instructions in a single clock cycle, the ATmega8535 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



3

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 4 shows the structure of the 32 general purpose working registers in the CPU.

Figure 4. AVR CPU General Purpose Working Registers

	7	0	Addr.	
	R0		0x00	
	R1		0x01	
	R2		0x02	
	…			
	R13		0x0D	
General	R14		0x0E	
Purpose	R15		0x0F	
Working	R16		0x10	
Registers	R17		0x11	
	…			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

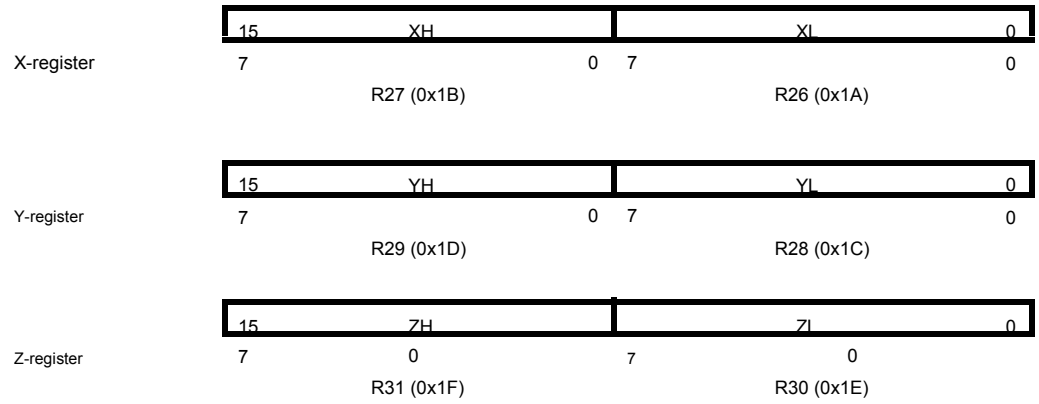
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 4, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-pointer Registers can be set to index any register in the file.

The X-register, Y-register, and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z are defined as described in Figure 5.

Figure 5. The X-, Y-, and Z-registers



In the different addressing modes, these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer.

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above 0x60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	**SP9**	**SP8**	**SPH**
	SP7	**SP6**	**SP5**	**SP4**	**SP3**	**SP2**	**SP1**	**SP0**	**SPL**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The AVR CPU is driven by the CPU clock clk_{CPU}, directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 6 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 6. The Parallel Instruction Fetches and Instruction Executions

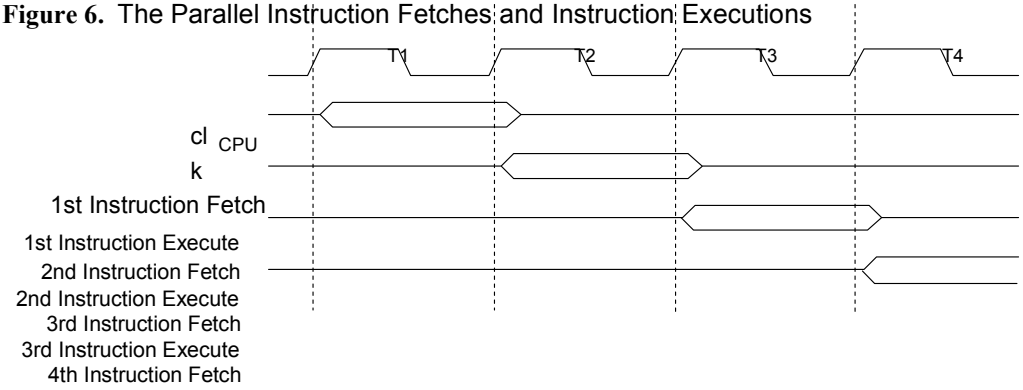
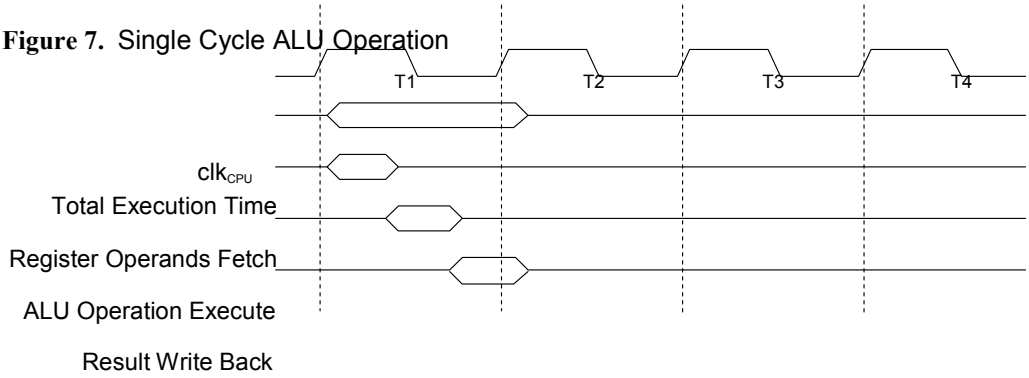


Figure 7 shows the internal timing concept for the Register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 7. Single Cycle ALU Operation



Reset and Interrupt Handling

The AVR provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate Program Vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt.

Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section “Memory Programming” on page 237 for details.

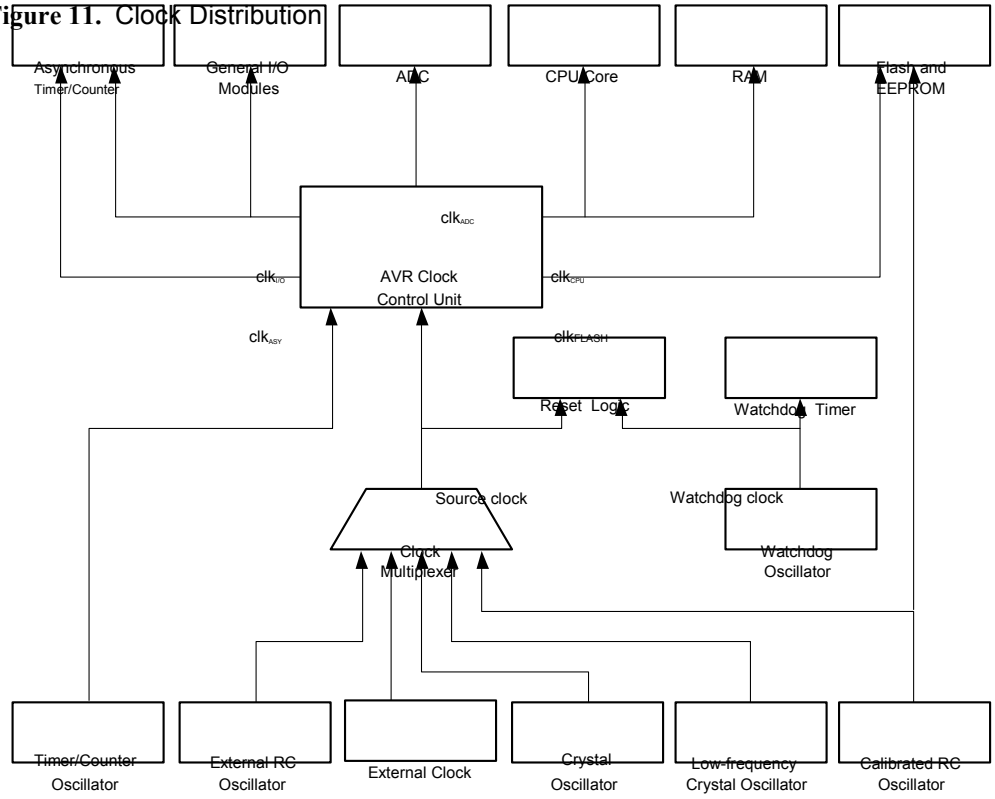
The lowest addresses in the program memory space are, by default, defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in “Interrupts” on page 46. The list also determines the priority levels of the different interrupts. The lower the address, the higher the priority level is. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the General Interrupt Control Register (GICR). Refer to “Interrupts” on page 46 for more information. The Reset Vector can

System Clock and Clock Options

Clock Systems and their Distribution

Figure 11 presents the principal clock systems in the AVR and their distribution. All of the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes, as described in “Power Management and Sleep Modes” on page 32. The clock systems are detailed below.

Figure 11. Clock Distribution



CPU Clock – clk_{CPU}

The CPU clock is routed to parts of the system concerned with operation of the AVR core. Examples of such modules are the General Purpose Register File, the Status Register and the data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

I/O Clock – $clk_{I/O}$

The I/O clock is used by the majority of the I/O modules, like Timer/Counters, SPI, and USART. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted. Also note that address recognition in the TWI module is carried out asynchronously when $clk_{I/O}$ is halted, enabling TWI address reception in all sleep modes.

Flash Clock – clk_{FLASH}

The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

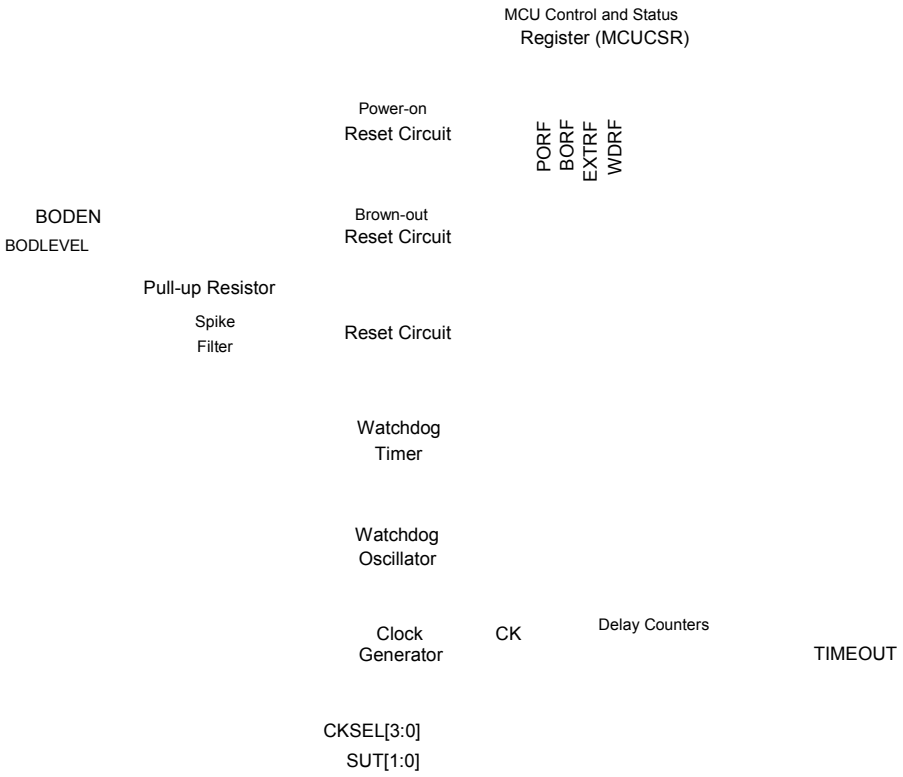


Table 15. Reset Characteristics

Symbol	Parameter	Condition	Min[1]	Typ[1]	Max[1]	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)			1.4	2.3	V
	Power-on Reset Threshold Voltage (falling)[2]			1.3	2.3	V
V_{RST}	RESET Pin Threshold Voltage		0.2		0.9	V
t_{RST}	Minimum pulse width on RESET Pin				1.5	µs
V_{BOT}	Brown-out Reset Threshold Voltage[3]	BODLEVEL = 1	2.5	2.7	2.9	V
		BODLEVEL = 0	3.6	4.0	4.2	
t_{BOD}	Minimum low voltage period for Brown-out Detection	BODLEVEL = 1		2		µs
		BODLEVEL = 0		2		µs
V_{HYST}	Brown-out Detector hysteresis			130		mV

Notes: 1. Values are guidelines only.

2. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling).

3. V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to $V_{CC} = V_{BOT}$ during the production test. This guarantees that a Brown-out Reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 1 for ATmega8535L and BODLEVEL = 0 for ATmega8535. BODLEVEL = 1 is not applicable for ATmega8535.

Power-on Reset

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The detection level is defined in Table 15. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the Start-up Reset, as well as to detect a failure in supply voltage.

A Power-on Reset (POR) circuit ensures that the device is reset from Power-on. Reaching the Power-on Reset threshold voltage invokes the delay counter, which determines how long the device is kept in RESET after V_{CC} rise. The RESET signal is activated again, without any delay, when V_{CC} decreases below the detection level.

Figure 16. MCU Start-up, RESET Tied to V_{CC}

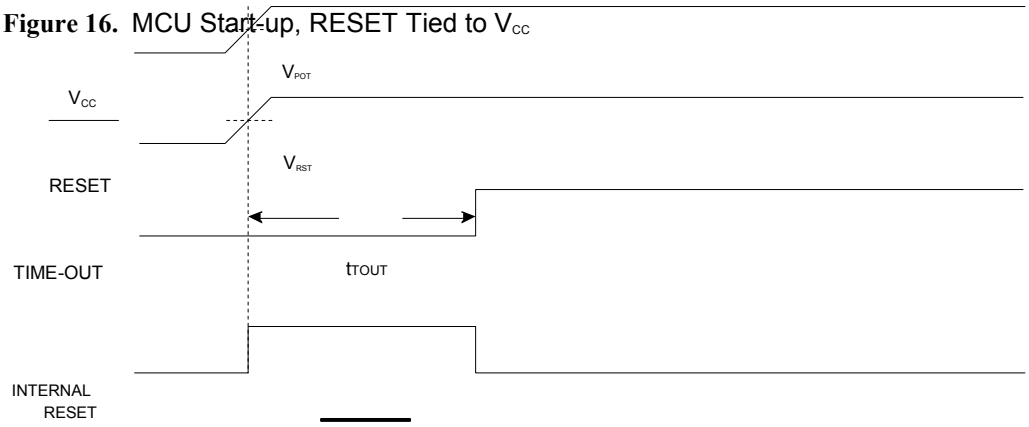
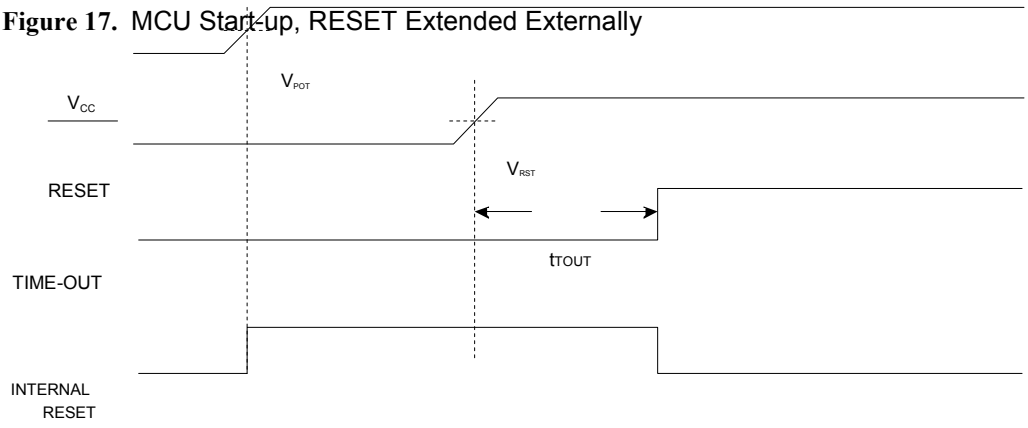


Figure 17. MCU Start-up, RESET Extended Externally

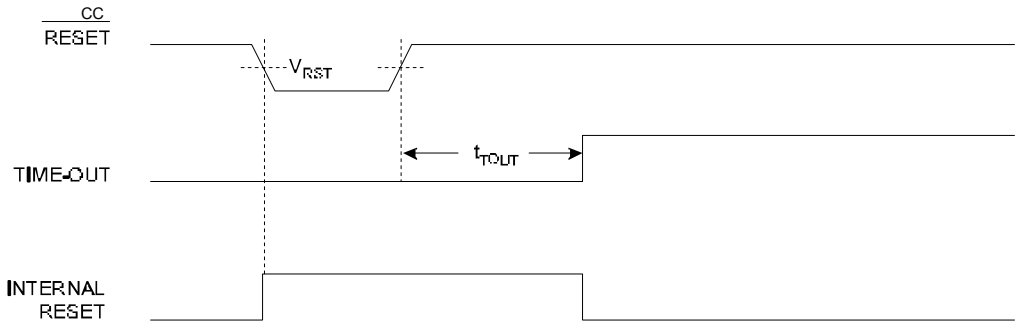


External Reset

An External Reset is generated by a low level on the RESET pin. Reset pulses longer than the minimum pulse width (see Table 15) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage – V_{RST} on its positive edge, the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

V

Figure 18. External Reset During Operation



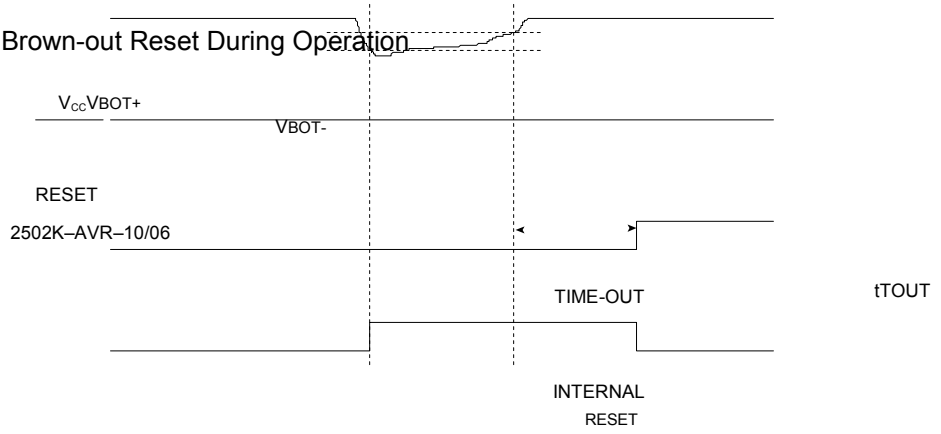
Brown-out Detection

ATmega8535 has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as $V_{BOT+} = V_{BOT} + V_{HYST}/2$ and $V_{BOT-} = V_{BOT} - V_{HYST}/2$.

The BOD circuit can be enabled/disabled by the fuse BODEN. When the BOD is enabled (BODEN programmed), and V_{CC} decreases to a value below the trigger level (V_{BOT-} in Figure 19), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level (V_{BOT+} in Figure 19), the delay counter starts the MCU after the time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} given in Table 15.

Figure 19. Brown-out Reset During Operation



February 2007

LM741
Single Operational Amplifier

Features

- Short Circuit Protection
- Excellent Temperature Stability
- Internal Frequency Compensation
- High Input Voltage Range
- Null of Offset

Description

The LM741 series are general purpose operational amplifiers. It is intended for a wide range of analog applications. The high gain and wide range of operating voltage provide superior performance in intergrator, summing amplifier, and general feedback applications..

8-DIP

1

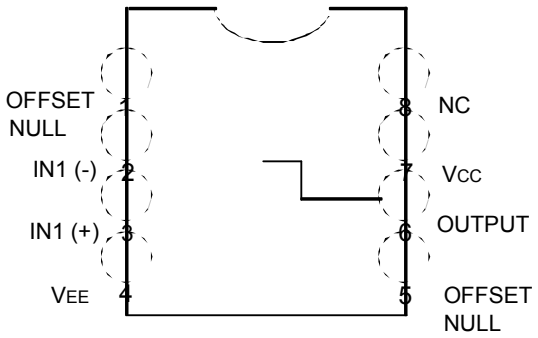
8-SOP

1

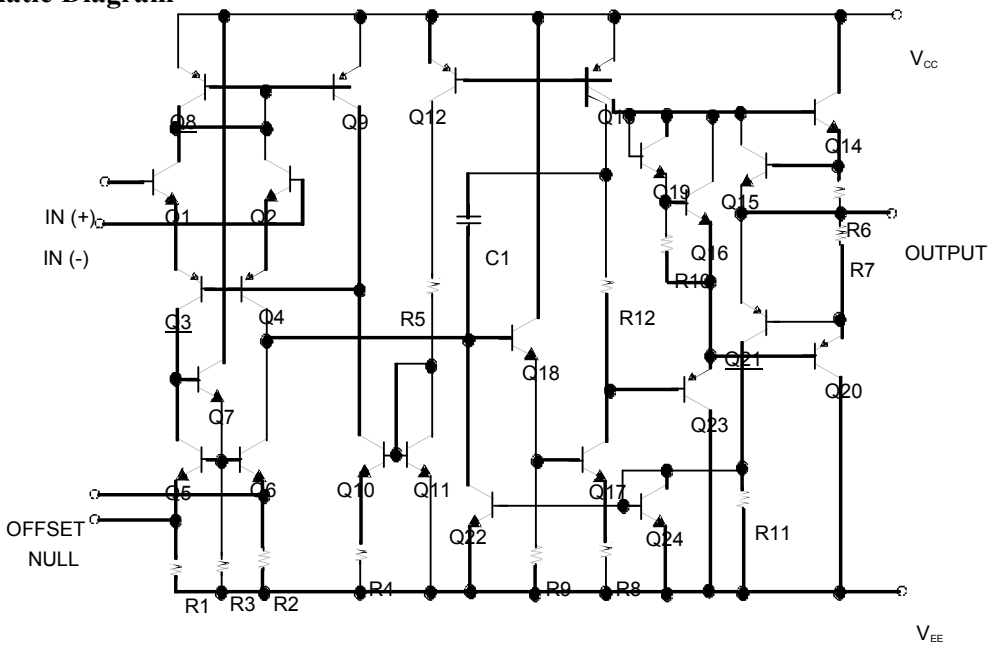
Ordering Information

Part Number	Operating Temp. Range	Pb-Free	Package	Packing Method	Marking Code
LM741CN	0 ~ +70°C	YES	8-DIP	Rail	LM741CN
LM741CM		YES	8-SOP	Rail	LM741CM
LM741CMX		YES	8-SOP	Tape & Reel	LM741CM

Internal Block Diagram



Schematic Diagram



Absolute Maximum Ratings

The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. T_A=25°C, unless otherwise specified.

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	±18	V
$V_{I(DIFF)}$	Differential Input Voltage	30	V
V_I	Input Voltage	±15	V
-	Output Short Circuit Duration	Indefinite	-
P_D	Power Dissipation	500	mW
T_{OPR}	Operating Temperature Range	0 ~ +70	°C
T_{STG}	Storage Temperature Range	-65 ~ +150	°C

Electrical Characteristics

(V_{CC}= 15V, V_{EE} = -15V, T_A = 25°C, unless otherwise specified)

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input Offset Voltage	V_{IO}	$R_S \leq 10k\Omega$		-	2.0	6.0	mV
		$R_S \leq 50\Omega$		-	-	-	
Input Offset Voltage Adjustment Range	$V_{IO(R)}$	V_{CC} = ±20V		-	±15	-	mV
Input Offset Current	I_{IO}			-	20	200	nA
Input Bias Current	I_{BIAS}			-	80	500	nA
Input Resistance (Note1)	R_I	V_{CC} = ±20V		0.3	2.0	-	MΩ
Input Voltage Range	$V_{I(R)}$			±12	±13	-	V
Large Signal Voltage Gain	G_V	$R_L \geq 2k\Omega$	V_{CC} = ±20V, $V_{O(P-P)}$ = ±15V	-	-	-	V/mV
			V_{CC} = ±15V, $V_{O(P-P)}$ = ±10V	20	200	-	
Output Short Circuit Current	I_{SC}			-	25	-	mA
Output Voltage Swing	$V_{O(P-P)}$	V_{CC} = ±20V	$R_L \geq 10k\Omega$	-	-	-	V
			$R_L \geq 2k\Omega$	-	-	-	
		V_{CC} = ±15V	$R_L \geq 10k\Omega$	±12	±14	-	
			$R_L \geq 2k\Omega$	±10	±13	-	
Common Mode Rejection Ratio	CMRR	$R_S \leq 10k\Omega$, V_{CM} = ±12V		70	90	-	dB
Power Supply Rejection Ratio	PSRR	$R_S \leq 50\Omega$, V_{CM} = ±12V		-	-	-	dB
		V_{ee} = ±15V to V_{ee} = ±15V		-	-	-	
		$R_S \leq 50\Omega$					
		V_{ee} = ±15V to V_{ee} = ±15V		77	96	-	
		$R_S \leq 10k\Omega$					
Transient Response Rise Time	T_R	Unity Gain		-	0.3	-	μs
Overshoot	OS			-	10	-	%
Bandwidth	BW	-		-	-	-	MHz
Slew Rate	SR	Unity Gain		-	0.5	-	V/μs
Supply Current	I_{CC}	R_L = ∞Ω		-	1.5	2.8	mA
Power Consumption		V_{CC} = ±20V		-	-	-	
	P_C	V_{CC} = ±15V		-	50	85	mW

Note:

1. Guaranteed by design.

Electrical Characteristics (Continued)

(0°C < T_A< 70 °C, V_{CC} = ±15V, unless otherwise specified)

The following specification apply over the range of 0°C ≤ T_A≤ +70°C for the LM741C							
Parameter	**Symbol**	**Conditions**		**Min.**	**Typ.**	**Max.**	**Unit**
Input Offset Voltage	V_{IO}	R_S≤ 50Ω		-	-	-	mV
		R_S ≤ 10kΩ		-	-	7.5	
Input Offset Voltage Drift	$\Delta V_{IO}/\Delta T$	-		-	-		µV/°C
Input Offset Current	I_{IO}	-		-	-	300	nA
Input Offset Current Drift	$\Delta I_{IO}/\Delta T$	-		-	-		nA/°C
Input Bias Current	I_{BIAS}	-		-	-	0.8	µA
Input Resistance (Note1)	R_I	V_{CC} = ±20V		-	-	-	MΩ
Input Voltage Range	$V_{I(R)}$	-		±12	±13	-	V
Output Voltage Swing	$V_{O(P-P)}$	V_{CC} =±20V	R_S≥ 10kΩ	-	-	-	V
			R_S≥ 2kΩ	-	-	-	
		V_{CC} =±15V	R_S≥ 10kΩ	±12	±14	-	
			R_S≥ 2kΩ	±10	±13	-	
Output Short Circuit Current	I_{SC}	-		10	-	40	mA
Common Mode Rejection Ratio	CMRR	R_S≤ 10kΩ, V_{CM} = ±12V		70	90	-	dB
Power Supply Rejection Ratio	PSRR	R_S≤ 50Ω, V_{CM} = ±12V		-	-	-	
		V_{CC} = ±20V to	R_S≤ 50Ω	-	-	-	dB
Large Signal Voltage Gain	G_V	±5V		77	96	-	
		R_S≥ 2kΩ	R_S≤ 10kΩ				
			V_{CC} = ±20V, $V_{O(P-P)}$ = ±15V	-	-	-	
			V_{CC} = ±15V, $V_{O(P.P)}$ = ±10V	15	-	-	V/mV
			V_{CC} = ±15V, $V_{O(P-P)}$ = ±2V	-	-	-	

Note :

1. Guaranteed by design.

Typical Performance Characteristics

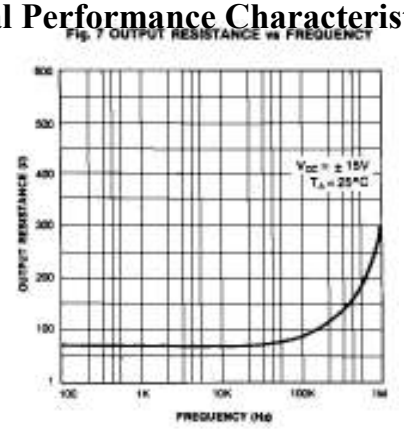


Figure 1. Output Resistance vs Frequency

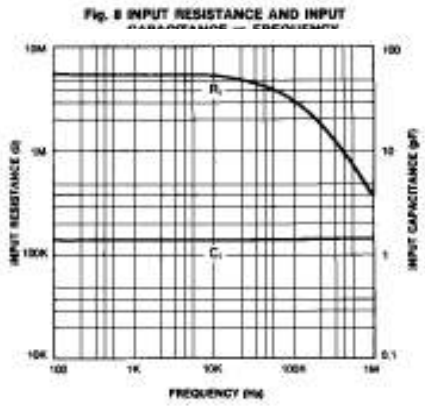


Figure 2. Input Resistance and Input Capacitance vs Frequency

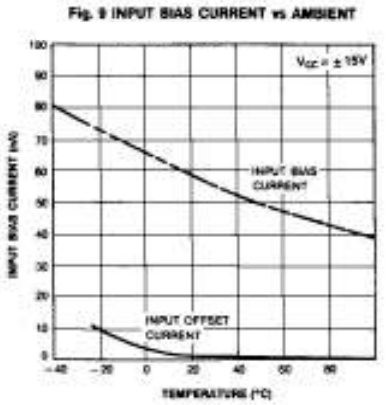


Figure 3. Input Bias Current vs Ambient Temperature

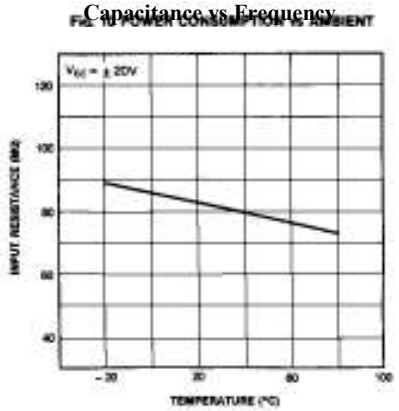


Figure 4. Power Consumption vs Ambient Temperature

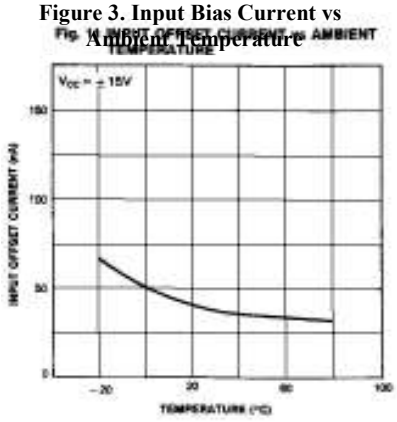


Figure 5. Input Offset Current vs Ambient Temperature

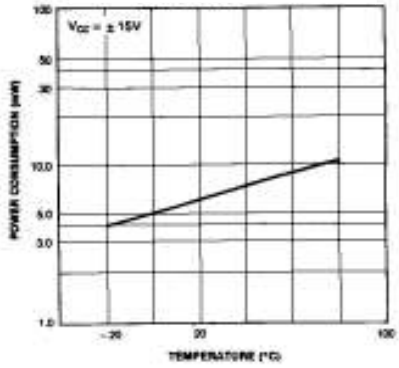


Figure 6. Input Resistance vs Ambient Temperature

Typical Performance Characteristics (Continued)

Fig. 13 NORMALIZED DC PARAMETERS

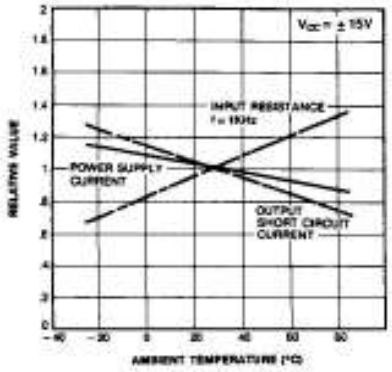


Figure 7. Normalized DC Parameters vs Ambient Temperature

Fig. 14 FREQUENCY CHARACTERISTICS

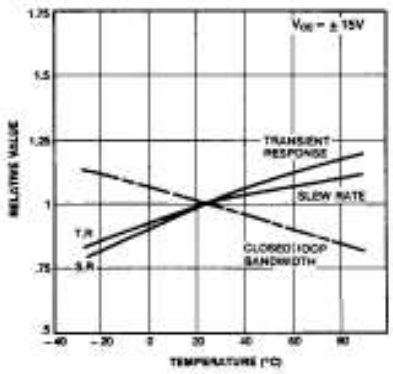


Figure 8. Frequency Characteristics vs Ambient Temperature

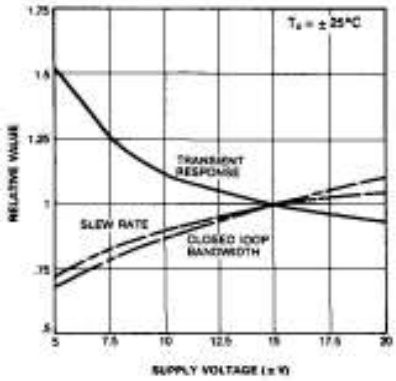


Figure 9. Frequency Characteristics vs Supply Voltage

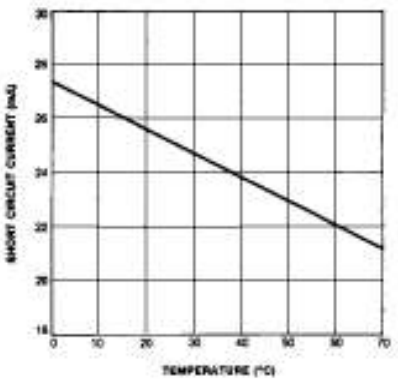


Figure 10. Output Short Circuit Current vs Ambient Temperature

Fig. 11 TRANSIENT RESPONSE

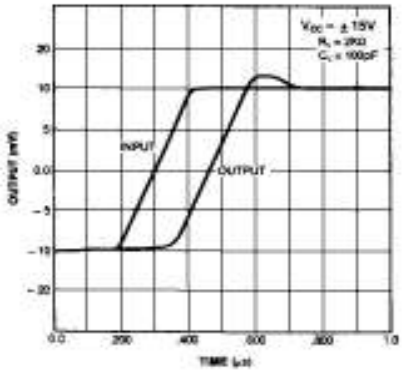


Figure 11. Transient Response

Fig. 12 COMMON-MODE REJECTION RATIO

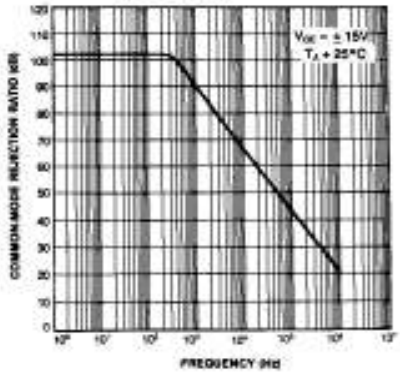


Figure 12. Common-Mode Rejection Ratio vs Frequency

Typical Performance Characteristics (Continued)

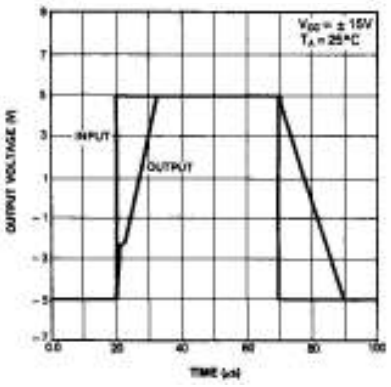


Figure 1. Voltage Follower Large Signal Pulse Response

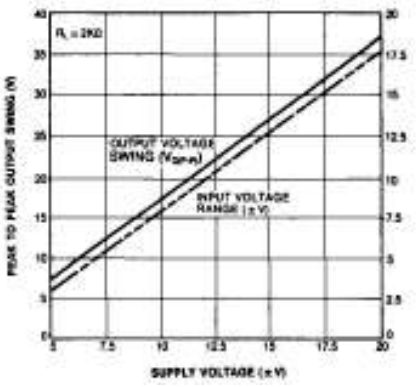


Figure 2. Output Swing and Input Range vs Supply Voltage

Single Supply Operation of the DAC0800 and DAC0802

National Semiconductor
Application Note 1525
Nick Gray
September 2006

1.0 Executive Summary

The ADC0800 and the ADC0802 are versatile 4 quadrant multiplying DACs (Digital-to-Analog Converters) with a current reference and complementary current outputs. The data sheet indicates a need for bipolar (positive and negative) power supply voltages, but a consideration of relative potentials allows these DACs to be used with positive only supplies, to include a single positive supply. This application note develops the positive only bias voltages needed to successfully operate the ADC0800 and the DAC0802 with a single supply voltage. The principles here also apply to the DAC0808, which is a two quadrant multiplying DAC with a single output.

The DAC0800 is functionally equivalent to the industry standard DAC-08 and the DAC0802 is equivalent to the industry standard DAC-08A. The DAC0808 is functionally equivalent to the industry standard MC1408.

Product Applicability:

DAC0800
DAC0802
DAC0808

2.0 Overview

The DAC0800, the DAC0802 and the DAC0808 seem to suffer from the apparent need for bipolar supplies. But, by setting the negative supply to ground and shifting all other voltages up by 5 Volts, these DACs will behave normally without a negative supply. The requirement is that all of the voltages associated with the DAC are correct relative to each other, and that the proper current levels are maintained.

This application note does not discuss the operation or specifications of the DACs to which this document applies. See the product data sheet for information relating to the operation of the product in question.

Reference to the DAC0800 herein also includes the DAC0802, which is just a more accurate version of the DAC0800. The principles here apply equally well to the DAC0808, except that pin 2 of the DAC0808 is normally ground rather than an output.

3.0 General Biasing Requirements

As long as the relative voltages are correct, the DAC0800 will behave normally. With this in mind, the potentials around the DAC0800 can be modified as indicated in *Table 1*.

The positive supply voltage, V+ at pin 13, must be at least 10V and no more than 30V more positive than V- at pin 3 for proper operation.

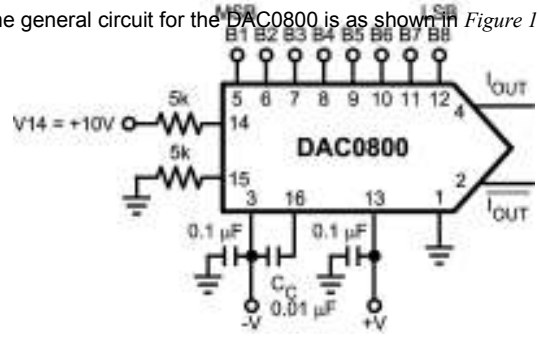
There will be no change to the capacitor connection to pin 16, but it should be noted that this capacitor should be connected directly to pin 3, then this connection should be connected to the negative supply voltage or grounded.

Pin 1, the Logic Control pin, controls the logic threshold of the digital inputs. The logic threshold of the digital input is about 1.35V above this pin 1 potential at room temperature and varies linearly between about 1.7V at -55˚C and about 0.95V at 125˚C. The assumption throughout this application note is 5 Volt TTL logic levels.

TABLE 1. DAC0800 Voltages and Currents

Pin	Function	Normal V or I (or Range)	New V or I (or Range)
1	Logic Control	0V	+5V
2	I_o	−10V to +18V	(V−) + 5V to +28V
3	V-	−15V to −5V	0V
4	I_o	−10V to +18V	(V−) +5 to +28
5	B1 (MSB)		
6	B2	0V to 0.8V	5.0V to 5.8V
7	B3	(logic low)	(logic low)
8	B4		
9	B5		
10	B6	2.4V to V+	7.4V to V+
11	B7	(logic high)	(logic high)
12	B8 (LSB)		
13	V+	+5V to +15V	0V to +20V
14	REF+	200 µA to 4 mA	200 µA to 4 mA
15	REF-	0V thru R	+5V thru R
16	COMP	cap	cap

The general circuit for the DAC0800 is as shown in *Figure 1*.



20207701

FIGURE 1. General Circuit for the DAC0800

The requirements for reference pins 14 and 15 are that they be at least 5 Volts above the negative supply pin 3, that the resistance in series with these two pins be of equal value and that the value of these two resistors be (V14 - V15) / I_{REF}, where V14 and V15 are the potentials to which the resistors at pins 14 and 15 are returned and I_{REF} is the

3.0 General Biasing Requirements

(Continued)

reference current required and is equal to the value of the maximum output currents at pins 4 and 2. The total of these two output currents is always equal to

$$I_{OUT} + I_{\overline{OUT}} = (2^n - 1)\, I_{REF} / 2^n$$

For 8 bits this becomes

$$I_{OUT} + I_{\overline{OUT}} = 255\, I_{REF} / 256$$

I_{OUT} increases with an increase in the digital code and/or the reference current and I_{OUT} decreases with an increase in the digital code and/or the reference current.

We see from *Figure 1* that

$$I_{REF} = (V14 - V15) / 5k$$

$$I_{REF} = (10V - 0V) / 5k = 2\ mA$$

4.0 Unipolar Supply Modification

Again, if we consider the relative circuit potentials and maintain proper current levels, we can make whatever circuit changes we need.

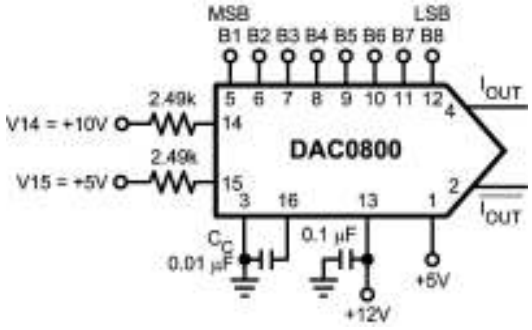
4.1 REFERENCE CONSIDERATIONS

The modified circuit is as shown in *Figure 2*. Here we find that

$$I_{REF} = (V14 - V15) / 2.49k$$

$$I_{REF} = (10V - 5V) / 2.49k = 2.008\ mA$$

and we have about the same reference current as for the general case.



20207702

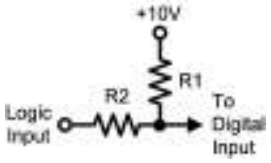
FIGURE 2. Positive only supply circuit for the DAC0800

4.2 OUTPUT BIASING

Another requirement of the DAC0800 is that output pins 2 and 4 must always be at least 5 V more positive than the potential at pin 3. Since pin 3 is grounded, these output pins must never go below +5V. The +28V maximum output voltage indicated in *Table 1* comes from the maximum voltage with respect to pin 3 that the output transistors can withstand without excessive leakage or breakdown.

4.3 INPUT LEVEL THRESHOLD

The logic inputs need to be level shifted up by 5V. This can be done with two resistors at each logic input pin, as shown in *Figure 3*.



20207703

FIGURE 3. Simple digital input level shifting circuit

For driving with 5V TTL devices, R1 and R2 of *Figure 3* are 2.65k and 3.3k, respectively. For driving with 3.3V TTL devices, R1 and R2 are 3.48k and 4.75k, respectively. These resistance values would change for bias voltages other than the +10V shown in *Figure 3*. If the bias voltage is +12V, R1 and R2 of are 5.83k and 4.75k, respectively, for 5V TTL devices and 5.62k and 4.75k, respectively, for driving with 3.3V TTL devices.

Tolerance requirements of both the resistors and of the bias voltage is 1%, so it is best to use a reference source for the bias voltage.

4.4 OUTPUT CONSIDERATIONS

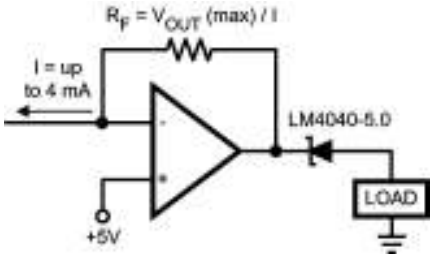
The DAC0800 has a current output. The data sheet indicates how to connect an amplifier to the outputs to derive a voltage output. Keep in mind that the DAC0800 outputs must never transition to a potential that is lower than 5V above its pin 3 voltage.

This means that an op-amp at the output must have its non-inverting input returned to a voltage that is it least +5 Volts, leading to a minimum op-amp output of +5V. In some cases this may be acceptable, but in most cases it is much more desirable to have a minimum output voltage of zero volts.

This minimum zero volts output can be obtained by adding a Zener diode in series with the output to subtract the bias potential from the output, but a Zener diode generally has a large tolerance and a large temperature coefficient. Using an LM4040-5.0 in place of a Zener will provide the same subtraction function, but with a lot better tolerance and lower temperature drift. An example circuit, where the DAC0800 output bias is 5V, is shown in *Figure 4*.

4.0 Unipolar Supply Modification

(Continued)



20207704

FIGURE 4. Current-to-voltage converter with offset correction.

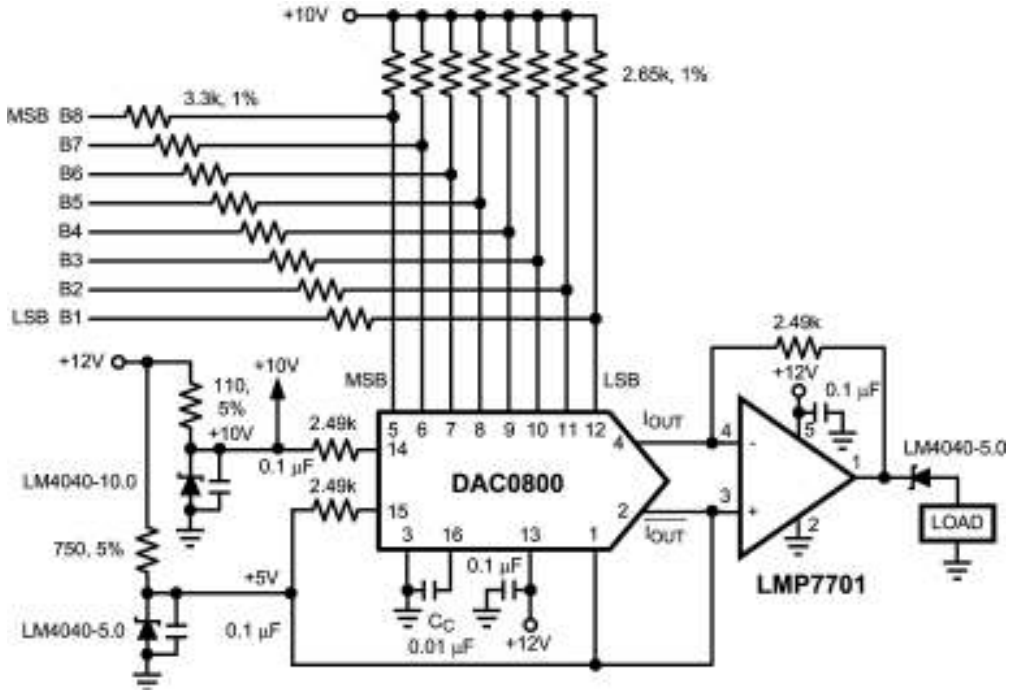
There are two things to note about the circuit of *Figure 4*. The supply voltage should be more positive than the maximum amplifier output, even if a rail-to-rail output amplifier is used. Otherwise there could be linearity problems at and near full scale.

Second, the maximum current through the LM4040-5.0 is 15 mA, which limits the load current to that value.

The third thing is that the LM4040-5.0 should have a minimum current of 100 µA at all times if it is to reliably subtract 5V from the amplifier output. When the amplifier output is 5V, there is insufficient current through the LM4040-5.0, resulting in an inaccurate load voltage at low load currents. That is, the load voltage can not be reduced to zero at low load currents. The minimum voltage will be a function of the load impedance and the individual LM4040-5.0. For this reason, this circuit may not be practical for applications where the load current can be reduced to a value below 100 microamps.

4.5 COMPLETE CIRCUIT EXAMPLE

Figure 5 gives a complete positive supply circuit solution to provide an output range or 0 to 5V. The Op-amp is a rail-to-rail output type to minimize its output error at very low output voltages. The positive supply voltage for the op-amp should be greater than its maximum output voltage.



20207705

FIGURE 5. A complete single supply DAC0800 design with 0V to 5V output.

5.0 Summary

The DAC0800 is a versatile DAC that has found many uses. The perceived drawback has been its need for a negative supply. However, because the important thing for any electrical component is the relative potentials and correct currents, it is possible to come up with a method to eliminate the need for a negative supply.