

## Register usage in Turbo (taken from Turbo source code)

Reg	Description	Name
A1	Maths stack pointer	maths
A4	Heap pointer	data
A5	Pseudo program pointer	thread
A6	Task base pointer	base
D0	QDOS Request code register	request
D5	Turbo error number	error
D6	(low word) BASIC line number	lineno

## Data areas in compiled programs

### Offsets from the start of the compiled program

#### CodeGen V5.00

\$00	W	Branch always to the start of program code.
\$02	L	CodeGen version - 5.00
\$06	W	\$4AFB marker word
\$08	W	string length of job name
\$0A		Bytes of job name
\$18		Copyright notice - 1986 Simon N Goodwin
.		
.		
.		
\$40	L	WHEN_ERROR 1 address
\$44	L	WHEN_ERROR 0 address
.		
.		
.		
\$5C		Start of program code

#### CodeGen V5.37

\$00	W	Branch always to another branch to the start of program code (offset \$48)
\$02	L	CodeGen version - 5.37
\$06	W	\$4AFB marker word
\$08	W	string length of job name
\$0A		Bytes of job name
\$18		Copyright notice - 2000 The Turbo Team
.		
.		
.		
\$48	W	Branch to the start of program code.
\$4A	L	WHEN_ERROR 1 address
\$4E	L	WHEN_ERROR 0 address
.		
.		
.		
\$88		Start of program code

A few samples of code routines used by the compiled SuperBASIC that appear to write to the program code area, rather than the data area.

The WHEN\_ERROR's pointing back to about \$4x from the start of the job. (see above)

It also looks like there is some channel data (channel definition blocks?) also stored in the program area.

Note that      A5 is program pointer for the compiled SuperBASIC  
                  A6 points to about 32K from the start of the job

Version 5.37 - WHEN\_ERROR 0

Code Index = 172

Checksum = 0010003258

00282680	201D	move.l	(a5)+, d0	.
00282682	45EE8056	lea	-\$7FAA(a6), a2	E..V
00282686	248D	move.l	a5, (a2)	\$.
00282688	4BF60800	lea	\$00(a6, d0.1), a5	K...
0028268C	4EEE8370	jmp	-\$7C90(a6)	N..p

Version 5.37 - WHEN\_ERROR 1

Code Index = 173

Checksum = 001000322C

002827B0	201D	move.l	(a5)+, d0	.
002827B2	45EE8052	lea	-\$7FAE(a6), a2	E..R
002827B6	248D	move.l	a5, (a2)	\$.
002827B8	4BF60800	lea	\$00(a6, d0.1), a5	K...
002827BC	4EEE8370	jmp	-\$7C90(a6)	N..p

Version 5.37 - END\_WHEN/CONTINUE

Code Index = 174

Checksum = 0012003FB3

0028279E	41EE8056	lea	-\$7FAA(a6), a0	A..V
002827A2	2410	move.l	(a0), d2	\$.
002827A4	08D00007	bset	#\$07, (a0)	....
002827A8	4EEE8B0E	jmp	-\$74F2(a6)	N...
002827AC	4EEE8370	jmp	-\$7C90(a6)	N..p

Version 5.37 - RETRY

Code Index = 175

Checksum = 001800583E

002828BC	242E805A	move.l	-\$7FA6(a6), d2	\$. .Z
002828C0	08AE00078052	bclr	#\$07, -\$7FAE(a6)	.....R
002828C6	08AE00078056	bclr	#\$07, -\$7FAA(a6)	.....V
002828CC	4EEE8B0E	jmp	-\$74F2(a6)	N...
002828D0	4EEE8370	jmp	-\$7C90(a6)	N..p

\*\*\* Channels \*\*\*

Version 5.37 - Check channel is open

Code Index = 180

Checksum = 00260136B7

00282696	3219	move.w	(a1)+,d1	2.
00282698	B27C0020	cmp.w	#\$0020,d1	. ..
0028269C	6504	bcs.s	\$002826A2	e.
0028269E	4EEE8AE8	jmp	-\$7518(a6)	N...
002826A2	C2FC0028	mulu	#\$0028,d1	...(
002826A6	45EE8ED8	lea	-\$7128(a6),a2	E...
002826AA	47F21004	lea	\$04(a2,d1.w),a3	G...
002826AE	248B	move.l	a3,(a2)	\$.
002826B0	4A13	tst.b	(a3)	J.
002826B2	6BEA	bmi.s	\$0028269E	k.
002826B4	422E8ED6	clr.b	-\$712A(a6)	B...
002826B8	4EEE8370	jmp	-\$7C90(a6)	N..p

Version 5.37 - RESTORE

Code Index = 165

Checksum = 000E00266B

00282930	2E1D	move.l	(a5)+,d7	..
00282932	45F67800	lea	\$00(a6,d7.l),a2	E.x.
00282936	2D4A8EBE	move.l	a2,-\$7142(a6)	-J..
0028293A	4EEE8370	jmp	-\$7C90(a6)	N..p

Version 5.37 - RETRY\_HERE

Code Index = 221

Checksum = 001A00C948

002825F0	41E9FF80	lea	-\$0080(a1),a0	A...
002825F4	B1CC	cmpa.l	a4,a0	..
002825F6	6404	bcc.s	\$002825FC	d.
002825F8	4EEE8AEC	jmp	-\$7514(a6)	N...
002825FC	45EE805A	lea	-\$7FA6(a6),a2	E..Z
00282600	24CD	move.l	a5,(a2)+	\$.
00282602	48D2D200	movem.l	a1/a4/a6-a7,(a2)	H...
00282606	4EEE8370	jmp	-\$7C90(a6)	N..p