/proc/cpuinfo Intel(R) Celeron(R) CPU 430 @ 1.80GHz
cache size: 512 KB

PARITY:	gcc -m32 -O<n> parity.c -o parity
	for ((i=0 ; i<11; i++)); do echo $i ; ./parity; done | pr -11 -l 20 -w 80
	ignorar medición 0, repetir columna si alguna medición se sale demasiado de la media

Zona para reproducir mediciones
recordar que se ignora medición 0

Optimización -O0	0	1	2	3	4	5	6	7	8	9	10	media
parity1 (lenguaje C - for)	176628	160382	160285	160500	162826	160644	143246	160448	147280	160669	163075	157936
parity2 (lenguaje C -while)	100620	100935	100856	98291	100163	99629	109236	99669	109484	98303	100459	101703
parity3 (asm - cuerpo for)	8086	8093	8110	8107	8092	8097	8095	8089	8102	8093	8106	8098
parity4 (asm - proc entero)	1769	1759	1761	1758	1810	1759	1758	1759	1759	1760	1760	1764

media	0	1	2	3	4	5	6	7
162841	157865	145205	167155	170813	165923	164911	168715	164298
104431	112718	112320	104823	103413	103572	100504	104208	99813
8439	8097	8091	8708	8092	8090	10952	8103	8080
1769	1758	1783	1768	1772	1774	1772	1762	1763

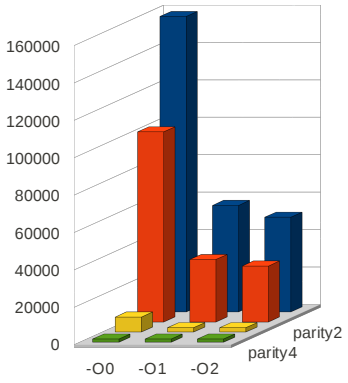
Optimización -O1	0	1	2	3	4	5	6	7	8	9	10	media
parity1 (lenguaje C - for)	69384	55707	60872	55790	55750	55691	58898	55740	55893	59067	55660	56907
parity2 (lenguaje C -while)	33196	33083	33088	33077	33183	35917	33077	33269	33075	33200	33184	33415
parity3 (asm - cuerpo for)	5642	2340	2446	2411	2444	2383	2441	2349	2383	2340	2347	2388
parity4 (asm - proc entero)	1758	1759	1760	1759	1757	1766	1768	1761	1761	1768	1760	1762

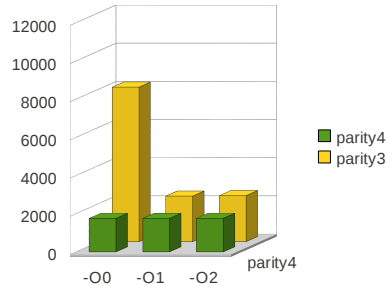
media	0	1	2	3	4	5	6	7
58527	69267	55746	58689	55649	59220	62530	55680	59099
33880	33205	33076	33079	33086	33175	37984	33074	34386
2409	2429	2411	2405	2379	2381	2401	2403	2464
1762	1758	1758	1758	1758	1760	1760	1760	1785

Optimización -O2	0	1	2	3	4	5	6	7	8	9	10	media
parity1 (lenguaje C - for)	67922	66828	51115	47425	47415	47628	50194	49985	47541	50567	47411	50611
parity2 (lenguaje C -while)	32654	29467	29467	29476	29466	31801	29495	29467	29523	29467	31854	29948
parity3 (asm - cuerpo for)	2484	2452	2444	2396	2371	2408	2463	2453	2391	2392	2454	2422
parity4 (asm - proc entero)	1761	1759	1760	1760	1761	1763	1762	1770	1763	1762	1766	1763

media	0	1	2	3	4	5	6	7
48980	61201	50939	47424	47460	47418	47563	52781	47403
31000	29495	29464	32285	29467	32160	29952	29467	29465
2409	2386	2406	2385	2441	2381	2443	2380	2390
1893	1758	1758	1758	1757	1756	1756	1757	1759

PARITY:	-O0	-O1	-O2
parity1	157936	56907	50611
parity2	101703	33415	29948
parity3	8098	2388	2422
parity4	1764	1762	1763





8	9	10
170954	164036	146401
100312	102774	112569
8064	8131	8076
1772	1762	1763

8	9	10
64309	55646	58704
34632	33076	33235
2450	2410	2386
1762	1761	1761

8	9	10
48373	49922	50513
30231	35505	32002
2416	2381	2467
2424	2442	1761

/proc/cpuinfo Intel(R) Celeron(R) CPU 430 @ 1.80GHz
cache size: 512 KB

POPCOUNT:	gcc -m32 -O<n> popcount.c -o popcount
	for ((i=0 ; i<11; i++)); do echo $i ; ./popcount; done | pr -11 -l 20 -w 80
	ignorar medición 0, repetir columna si alguna medición se sale demasiado de la media

Zona para reproducir mediciones
recordar que se ignora medición 0

Optimización -O0	0	1	2	3	4	5	6	7	8	9	10	media
popcount1 (lenguaje C - for)	171421	158527	158226	161033	158007	158336	158129	160819	160532	158027	141687	157332
popcount2 (lenguaje C - while)	100421	99589	96719	96542	96581	96642	99440	96553	96840	96657	104629	98019
popcount3 (asm adc- cuerpo for)	29943	29943	29917	33274	29977	29940	29974	29942	29936	29942	32691	30554
popcount4 (l.C-CS:APP 3.49-32b)	55112	54338	54268	55221	55084	54265	55178	55017	55553	55081	49234	54324
popcount5 (SSE3asm- cuerpo for)	2569	2566	2567	2560	2565	2579	2567	2576	2572	2567	2571	2569

media	0	1	2	3	4	5	6
157534	156386	161163	145796	163778	165062	158088	158617
102049	104651	104595	115533	96524	103009	101295	102172
30922	32867	29947	31768	32126	35571	29945	30053
54657	49238	55562	53661	55033	59707	55019	54268
2570	2561	2572	2574	2564	2562	2572	2569

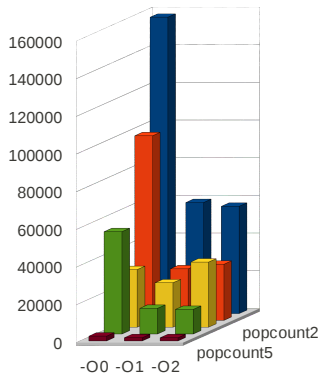
Optimización -O1	0	1	2	3	4	5	6	7	8	9	10	media
popcount1 (lenguaje C - for)	73247	57540	57690	55644	63594	58963	60271	59937	56364	60879	58287	58917
popcount2 (lenguaje C - while)	27497	27356	27363	27364	27368	27363	27395	27365	27532	27364	27365	27384
popcount3 (asm adc- cuerpo for)	23244	23243	23231	26486	23235	23236	23261	23371	23309	23235	23235	23584
popcount4 (l.C-CS:APP 3.49-32b)	13612	13626	13565	13647	13901	13738	13563	13572	13610	13562	13581	13637
popcount5 (SSE3asm- cuerpo for)	1703	1703	1705	1704	1720	1721	1703	1705	1705	1708	1707	1708

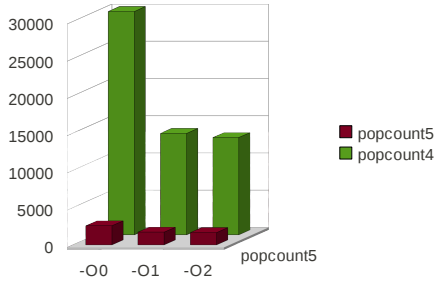
media	0	1	2	3	4	5	6
60167	77127	62616	62816	58504	62189	56606	59156
27953	27366	27364	27365	27576	30960	27366	29420
24490	23237	23231	23270	23236	23254	26294	28630
13678	13611	13558	13559	13579	13654	13590	14463
1763	1707	1698	1701	1699	1700	1703	2306

Optimización -O2	0	1	2	3	4	5	6	7	8	9	10	media
popcount1 (lenguaje C - for)	73676	56371	56231	57181	58685	56212	56527	56691	58997	56480	56267	56964
popcount2 (lenguaje C - while)	29765	29716	29659	31529	29655	29650	29758	29648	29690	29652	29650	29861
popcount3 (asm adc- cuerpo for)	34226	34207	34207	34207	34323	34677	37577	34340	34206	34220	34212	34618
popcount4 (l.C-CS:APP 3.49-32b)	13067	13061	13062	13067	13068	13139	13066	13068	13072	13065	13066	13073
popcount5 (SSE3asm- cuerpo for)	1657	1656	1656	1645	1644	1642	1645	1656	1640	1641	1643	1647

media	0	1	2	3	4	5	6
58723	73156	58692	59849	56342	59024	56232	57557
30452	29652	29655	29654	29650	29651	29787	31675
35245	34205	34210	34212	34222	34204	34204	36680
13739	13066	13069	13064	16634	13067	13067	14349
1642	1637	1637	1641	1640	1640	1642	1644

POPCOUNT:	-O0	-O1	-O2
popcount1	157332	58917	56964
popcount2	98019	27384	29861
popcount3	30554	23584	34618
popcount4	54324	13637	13073
popcount5	2569	1708	1647





7	8	9	10
138955	164637	158221	161025
104921	96658	99289	96497
29945	29943	29975	29942
49370	54250	54460	55235
2570	2574	2571	2571

7	8	9	10
56905	59645	60965	62272
27402	27357	27364	27357
23271	26805	23235	23674
13640	13554	13601	13582
1703	1710	1703	1702

7	8	9	10
60571	58663	59989	60311
31535	29653	31539	31716
34206	36885	36776	36848
13064	13100	13681	14293
1641	1641	1648	1644

/proc/cpuinfo Intel(R) Celeron(R) CPU 430 @ 1.80GHz
cache size: 512 KB

MATMULT:	gcc -m32 -O<n> matmult.c -o matmult
	for ((i=0 ; i<11; i++)); do echo $i ; ./matmult; done | pr -11 -l 20 -w 80
	ignorar medición 0, repetir columna si alguna medición se sale demasiado de la media

Zona para reproducir mediciones
recordar que se ignora medición 0

Optimización -O0	0	1	2	3	4	5	6	7	8	9	10	media
matmult1 (leng. C - index)	442862	425696	450042	395227	372941	406027	404296	348354	422222	379390	346663	395086
matmult2 (leng. C - pntrs)	281840	279354	284975	251571	248614	263344	267743	235364	283927	255740	240663	261130
matmult3 (copycol -rowmjr)	106488	106284	105809	110004	105593	105881	105760	106224	108461	105997	105595	106561
matmult4 (SSE3asm-pmuludq)	31968	31320	31687	32015	31179	31024	33969	30977	31255	30941	31030	31540

media	0	1	2	3	4	5	6
422933	410018	425799	415680	438627	415942	407172	435494
281941	259577	284873	277181	301921	270981	277631	287161
107147	109017	108398	111057	109682	106104	105392	106272
31962	31575	31267	31495	31034	37962	30950	31677

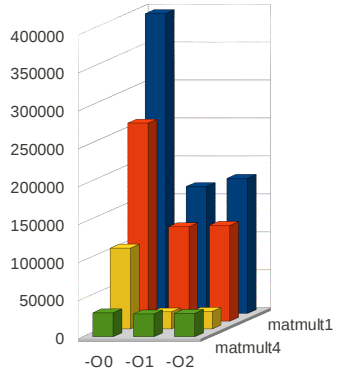
Optimización -O1	0	1	2	3	4	5	6	7	8	9	10	media
matmult1 (leng. C - index)	188503	190036	165220	165838	158681	162246	180192	139826	169358	174919	166758	167307
matmult2 (leng. C - pntrs)	128655	150064	135056	113355	117388	121366	142223	92335	118635	127599	126500	124452
matmult3 (copycol -rowmjr)	22771	22531	22764	22560	22341	22698	22288	22689	22355	22482	22389	22510
matmult4 (SSE3asm-pmuludq)	30284	30058	30187	30168	31383	30272	29939	30289	29950	30065	30338	30265

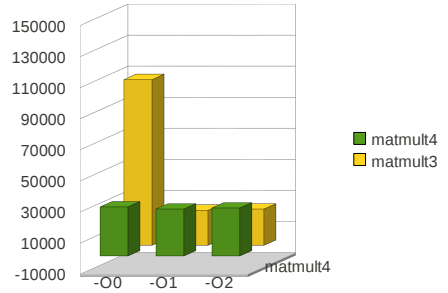
media	0	1	2	3	4	5	6
161230	157588	171556	172028	157792	150550	147406	156572
116343	107669	122579	118074	125906	112144	107134	109345
23487	22279	22699	26011	22653	22523	22573	22729
31016	29896	34396	30151	30279	30173	30141	30345

Optimización -O2	0	1	2	3	4	5	6	7	8	9	10	media
matmult1 (leng. C - index)	181059	239954	212272	169362	164430	160845	161116	159575	182320	199526	131611	178101
matmult2 (leng. C - pntrs)	136508	153353	145206	115180	126929	125197	117170	114646	124879	143730	97565	126386
matmult3 (copycol -rowmjr)	22858	22694	25844	23120	22864	23232	24773	22668	22602	22859	22845	23350
matmult4 (SSE3asm-pmuludq)	30462	30258	30825	30527	30153	30413	32309	30216	30066	30324	33555	30865

media	0	1	2	3	4	5	6
153082	186368	182617	170735	152267	100703	150215	115695
109367	112888	130259	115619	99975	75315	111638	93119
23287	26675	22748	22929	24131	22753	24467	24171
31242	30944	30167	30340	35432	30208	31955	32868

MATMULT:	-O0	-O1	-O2
matmult1	395086	167307	178101
matmult2	261130	124452	126386
matmult3	106561	22510	23350
matmult4	31540	30265	30865





7	8	9	10
455604	401801	347526	485680
304995	258879	235455	320334
105851	106600	106150	105964
30756	31527	31302	31648

7	8	9	10
179163	127792	188593	160851
137850	80460	123961	125975
22703	24073	26229	22672
33904	30194	30339	30234

7	8	9	10
178094	159063	138949	182480
120405	119744	96957	130640
22664	23170	22895	22944
30146	30609	30302	30391

/proc/cpuinfo Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz
cache size: 6144 KB

MATMULT:	gcc -m32 -O<n> matmult.c -o matmult
	for ((i=0 ; i<11; i++)); do echo $i ; ./matmult; done | pr -11 -l 20 -w 80
	ignorar medición 0, repetir columna si alguna medición se sale demasiado de la media

Zona para reproducir mediciones recordar que se ignora medición 0

Optimización -O0	0	1	2	3	4	5	6	7	8	9	10	media
matmult1 (leng. C - index)	90414	69214	73536	73214	69625	73295	72417	74659	71968	71588	73225	72274
matmult2 (leng. C - pntrs)	45578	48969	45299	45811	46359	46528	46980	46619	46558	46535	48487	46815
matmult3 (copycol -rowmjr)	36113	37073	36527	36179	37143	36410	38055	36567	38913	37431	37644	37194
matmult4 (SSE3asm-pmuludq)	8922	9068	8549	9000	8270	8597	8634	8681	8668	8652	8623	8674

media	0	1	2	3	4	5	6
73090	94677	68963	75444	71794	72336	73700	73181
46458	46684	45671	45508	46485	45551	46303	47100
37696	37054	36870	37880	36896	37640	38148	39468
8818	8954	8974	8509	8536	8348	8426	9676

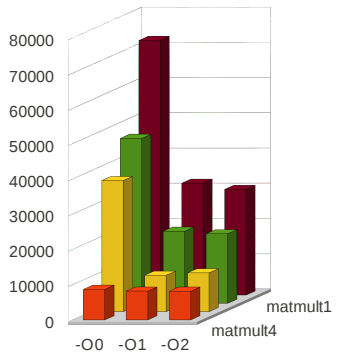
Optimización -O1	0	1	2	3	4	5	6	7	8	9	10	media
matmult1 (leng. C - index)	47445	32249	31971	32003	31566	31741	31501	31967	31387	31764	31334	31748
matmult2 (leng. C - pntrs)	20932	21556	20101	18756	22530	20735	20123	18458	17746	20165	24689	20486
matmult3 (copycol -rowmjr)	10877	10416	10811	10319	9925	10239	10232	10270	10057	10748	10225	10324
matmult4 (SSE3asm-pmuludq)	8384	8141	8122	8855	8092	7743	8023	7641	8119	8874	7975	8159

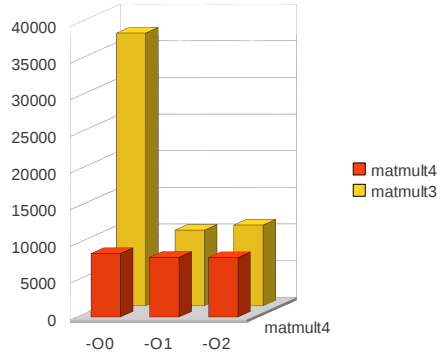
media	0	1	2	3	4	5	6
32139	51694	31555	32981	31816	33175	31244	34003
20220	21590	21210	20064	19286	21214	20178	22087
10165	10940	10026	9885	10562	10476	10348	10113
8422	7934	8461	8066	8079	7872	7753	8027

Optimización -O2	0	1	2	3	4	5	6	7	8	9	10	media
matmult1 (leng. C - index)	48896	32297	29428	31007	29785	28994	29677	30145	29201	29733	29754	30002
matmult2 (leng. C - pntrs)	21954	19918	20328	21112	17514	17816	16402	20509	21811	19777	22351	19754
matmult3 (copycol -rowmjr)	11363	10961	10991	10414	11297	10760	11590	11048	10959	11530	10550	11010
matmult4 (SSE3asm-pmuludq)	7694	8124	7826	8030	8115	8017	8171	8408	7971	7934	8773	8137

media	0	1	2	3	4	5	6
29696	54845	29558	29302	29414	29906	30144	29545
19832	18541	17263	19702	18681	22006	20075	20275
10966	10379	11653	10819	10766	10329	11002	10736
8240	8172	7747	8214	8144	8085	7938	7969

MATMULT:	-O0	-O1	-O2
matmult1	72274	31748	30002
matmult2	46815	20486	19754
matmult3	37194	10324	11010
matmult4	8674	8159	8137





7	8	9	10
73086	75466	75349	71581
48493	46972	46662	45836
37738	36944	37380	37992
8253	8953	9335	9168

7	8	9	10
31494	30788	31517	32812
21553	17722	20140	18747
9917	10058	10313	9956
8583	8176	8195	11012

7	8	9	10
29373	29420	30468	29825
20103	19197	21950	19068
11761	10396	10696	11506
7884	9551	8633	8234