Q: how to explain the "-33%" for li benchmark

- The only one **expands the cache line size**

- 99.9% hit rate and low bandwidth: very limited need of memory accessing

-

bench	Traditional cache arch		PaLM	
	organization	hit ratio	organization	hit ratio
go	8K/8/2	95	2K/4/2, 6K/8/2	95
compress	8K/128/2	99	2K/8/2, 6K/128/2	99
li	8K/32/2	99.9	2K/4/2, 6K/128/2	99.9
madd	512/8/2	83	256/4/2, 256/8/2	94
sor	8K/64/2	99	2K/32/2, 6K/64/2	99
vocoder	512/8/2	99	256/4/2, 256/8/2	99

Table 2. The traditional and customized local memory architectures and hit ratios.

bench	traditional cache arch		customized cache arch		% power decrease
	bandwidth	power	bandwidth	power	
go	0.32	0.832	0.26	0.67	23
compress	3.67	9.54	3.11	8.08	18
li	0.0156	0.04	0.0224	0.06	-33
madd	2.5	6.5	1.41	3.66	77
sor	0.31	0.806	0.19	0.49	63
vocoder	0.024	0.062	0.018	0.047	31
				average	30

Table 3. The bandwidth and power reductions obtained by our Local Memory Customization Algorithm.

Q: why there is a drop of hr for large line size

- Cache total size fixed (8KB)

- Larger line size ->

- fewer sets ->

- frequent replacement->
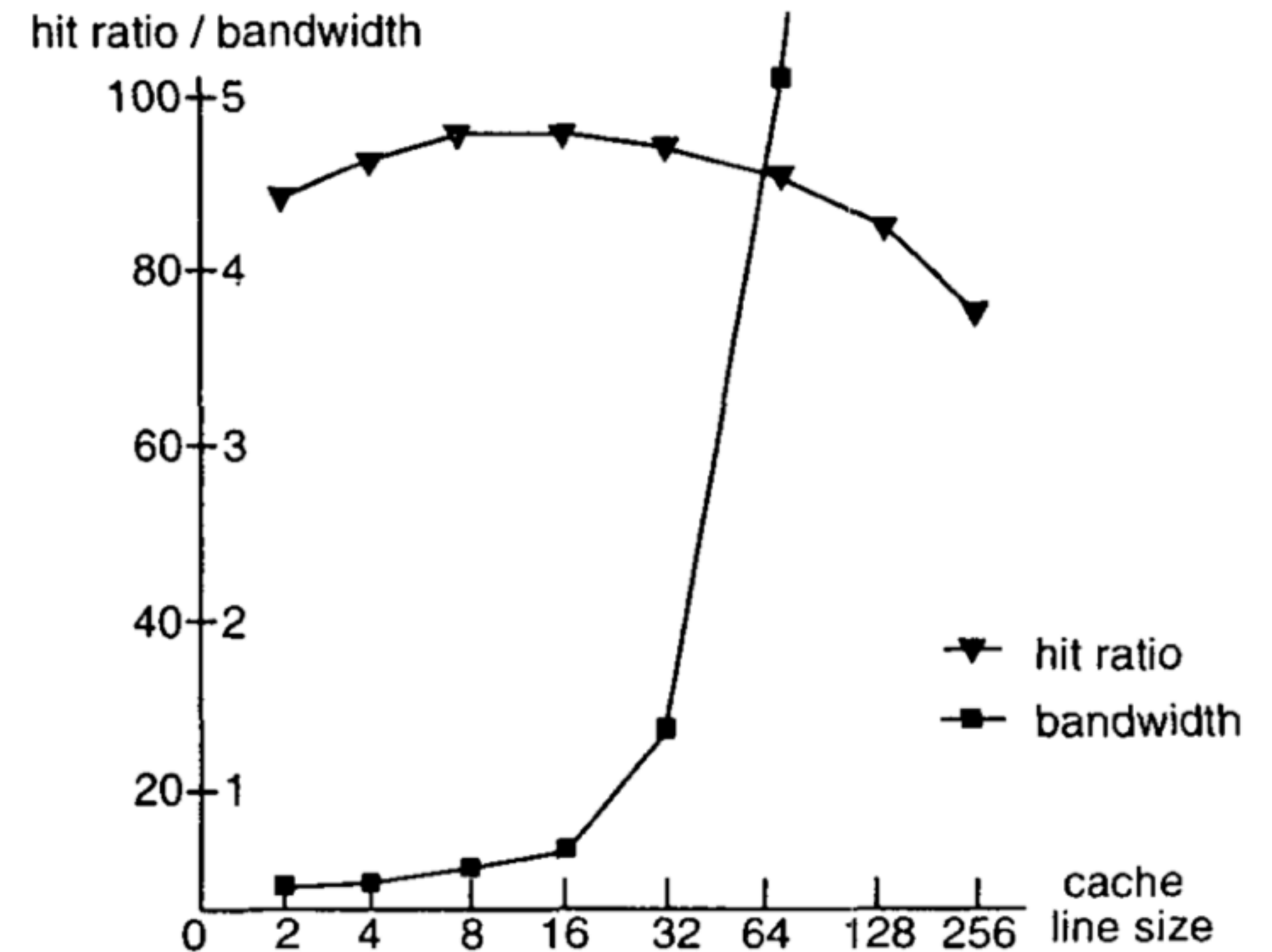
- Low hit rate

Figure 2. The cache hit ratio and memory bandwidth variation with the cache line size for the go benchmark