

Intro. to Computer Architecture

Daniel Page

Department of Computer Science,
University Of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB. UK.
(csdsp@bristol.ac.uk)

January 9, 2018

Keep in mind there are *two* PDFs available (of which this is the latter):

1. a PDF of examinable material used as lecture slides, and
2. a PDF of non-examinable, extra material:
 - ▶ the associated notes page may be pre-populated with extra, written explanation of material covered in lecture(s), plus
 - ▶ anything with a “grey’ed out” header/footer represents extra material which is useful and/or interesting but out of scope (and hence not covered).

Notes:

Notes:

- ▶ **Goal:** implement a design(s) for integer multiplication, i.e.,
 - ▶ accepts n -bit
 - ▶ y , the **multiplier** that does the multiplying, and
 - ▶ x , the **multiplicand** that is multiplied
 - as input, and
 - ▶ produces $2n$ -bit **product** $r = y \cdot x$ as output

which

1. functions correctly, *and*
2. is efficient (wrt. whatever metrics are deemed important).

Notes:

Notes:

INSTRUCTIONS SET DEFINITION, 6.1.1

6.1.1.1 Unaligned Multiplier

Field	Size (bits)	Field	Size (bits)	Field	Size (bits)
Op	4	Op2	4	Op3	4
Op4	4	Op5	4	Op6	4
Op7	4	Op8	4	Op9	4
Op10	4	Op11	4	Op12	4
Op13	4	Op14	4	Op15	4
Op16	4	Op17	4	Op18	4
Op19	4	Op20	4	Op21	4
Op22	4	Op23	4	Op24	4
Op25	4	Op26	4	Op27	4
Op28	4	Op29	4	Op30	4
Op31	4	Op32	4	Op33	4
Op34	4	Op35	4	Op36	4
Op37	4	Op38	4	Op39	4
Op40	4	Op41	4	Op42	4
Op43	4	Op44	4	Op45	4
Op46	4	Op47	4	Op48	4
Op49	4	Op50	4	Op51	4
Op52	4	Op53	4	Op54	4
Op55	4	Op56	4	Op57	4
Op58	4	Op59	4	Op60	4
Op61	4	Op62	4	Op63	4
Op64	4	Op65	4	Op66	4
Op67	4	Op68	4	Op69	4
Op70	4	Op71	4	Op72	4
Op73	4	Op74	4	Op75	4
Op76	4	Op77	4	Op78	4
Op79	4	Op80	4	Op81	4
Op82	4	Op83	4	Op84	4
Op85	4	Op86	4	Op87	4
Op88	4	Op89	4	Op90	4
Op91	4	Op92	4	Op93	4
Op94	4	Op95	4	Op96	4
Op97	4	Op98	4	Op99	4
Op100	4	Op101	4	Op102	4
Op103	4	Op104	4	Op105	4
Op106	4	Op107	4	Op108	4
Op109	4	Op110	4	Op111	4
Op112	4	Op113	4	Op114	4
Op115	4	Op116	4	Op117	4
Op118	4	Op119	4	Op120	4
Op121	4	Op122	4	Op123	4
Op124	4	Op125	4	Op126	4
Op127	4	Op128	4	Op129	4
Op130	4	Op131	4	Op132	4
Op133	4	Op134	4	Op135	4
Op136	4	Op137	4	Op138	4
Op139	4	Op140	4	Op141	4
Op142	4	Op143	4	Op144	4
Op145	4	Op146	4	Op147	4
Op148	4	Op149	4	Op150	4
Op151	4	Op152	4	Op153	4
Op154	4	Op155	4	Op156	4
Op157	4	Op158	4	Op159	4
Op160	4	Op161	4	Op162	4
Op163	4	Op164	4	Op165	4
Op166	4	Op167	4	Op168	4
Op169	4	Op170	4	Op171	4
Op172	4	Op173	4	Op174	4
Op175	4	Op176	4	Op177	4
Op178	4	Op179	4	Op180	4
Op181	4	Op182	4	Op183	4
Op184	4	Op185	4	Op186	4
Op187	4	Op188	4	Op189	4
Op190	4	Op191	4	Op192	4
Op193	4	Op194	4	Op195	4
Op196	4	Op197	4	Op198	4
Op199	4	Op200	4	Op201	4
Op202	4	Op203	4	Op204	4
Op205	4	Op206	4	Op207	4
Op208	4	Op209	4	Op210	4
Op211	4	Op212	4	Op213	4
Op214	4	Op215	4	Op216	4
Op217	4	Op218	4	Op219	4
Op220	4	Op221	4	Op222	4
Op223	4	Op224	4	Op225	4
Op226	4	Op227	4	Op228	4
Op229	4	Op230	4	Op231	4
Op232	4	Op233	4	Op234	4
Op235	4	Op236	4	Op237	4
Op238	4	Op239	4	Op240	4
Op241	4	Op242	4	Op243	4
Op244	4	Op245	4	Op246	4
Op247	4	Op248	4	Op249	4
Op250	4	Op251	4	Op252	4
Op253	4	Op254	4	Op255	4
Op256	4	Op257	4	Op258	4
Op259	4	Op260	4	Op261	4
Op262	4	Op263	4	Op264	4
Op265	4	Op266	4	Op267	4
Op268	4	Op269	4	Op270	4
Op271	4	Op272	4	Op273	4
Op274	4	Op275	4	Op276	4
Op277	4	Op278	4	Op279	4
Op280	4	Op281	4	Op282	4
Op283	4	Op284	4	Op285	4
Op286	4	Op287	4	Op288	4
Op289	4	Op290	4	Op291	4
Op292	4	Op293	4	Op294	4
Op295	4	Op296	4	Op297	4
Op298	4	Op299	4	Op300	4
Op301	4	Op302	4	Op303	4
Op304	4	Op305	4	Op306	4
Op307	4	Op308	4	Op309	4
Op310	4	Op311	4	Op312	4
Op313	4	Op314	4	Op315	4
Op316	4	Op317	4	Op318	4
Op319	4	Op320	4	Op321	4
Op322	4	Op323	4	Op324	4
Op325	4	Op326	4	Op327	4
Op328	4	Op329	4	Op330	4
Op331	4	Op332	4	Op333	4
Op334	4	Op335	4	Op336	4
Op337	4	Op338	4	Op339	4
Op340	4	Op341	4	Op342	4
Op343	4	Op344	4	Op345	4
Op346	4	Op347	4	Op348	4
Op349	4	Op350	4	Op351	4
Op352	4	Op353	4	Op354	4
Op355	4	Op356	4	Op357	4
Op358	4	Op359	4	Op360	4
Op361	4	Op362	4	Op363	4
Op364	4	Op365	4	Op366	4
Op367	4	Op368	4	Op369	4
Op370	4	Op371	4	Op372	4
Op373	4	Op374	4	Op375	4
Op376	4	Op377	4	Op378	4
Op379	4	Op380	4	Op381	4
Op382	4	Op383	4	Op384	4
Op385	4	Op386	4	Op387	4
Op388	4	Op389	4	Op390	4
Op391	4	Op392	4	Op393	4
Op394	4	Op395	4	Op396	4
Op397	4	Op398	4	Op399	4
Op400	4	Op401	4	Op402	4
Op403	4	Op404	4	Op405	4
Op406	4	Op407	4	Op408	4
Op409	4	Op410	4	Op411	4
Op412	4	Op413	4	Op414	4
Op415	4	Op416	4	Op417	4
Op418	4	Op419	4	Op420	4
Op421	4	Op422	4	Op423	4
Op424	4	Op425	4	Op426	4
Op427	4	Op428	4	Op429	4
Op430	4	Op431	4	Op432	4
Op433	4	Op434	4	Op435	4
Op436	4	Op437	4	Op438	4
Op439	4	Op440	4	Op441	4
Op442	4	Op443	4	Op444	4
Op445	4	Op446	4	Op447	4
Op448	4	Op449	4	Op450	4
Op451	4	Op452	4	Op453	4
Op454	4	Op455	4	Op456	4
Op457	4	Op458	4	Op459	4
Op460	4	Op461	4	Op462	4
Op463	4	Op464	4	Op465	4
Op466	4	Op467	4	Op468	4
Op469	4	Op470	4	Op471	4
Op472	4	Op473	4	Op474	4
Op475	4	Op476	4	Op477	4
Op478	4	Op479	4	Op480	4
Op481	4	Op482	4	Op483	4
Op484	4	Op485	4	Op486	4
Op487	4	Op488	4	Op489	4
Op490	4	Op491	4	Op492	4
Op493	4	Op494	4	Op495	4
Op496	4	Op497	4	Op498	4
Op499	4	Op500	4	Op501	4
Op502	4	Op503	4	Op504	4
Op505	4	Op506	4	Op507	4
Op508	4	Op509	4	Op510	4
Op511	4	Op512	4	Op513	4
Op514	4	Op515	4	Op516	4
Op517	4	Op518	4	Op519	4
Op520	4	Op521	4	Op522	4
Op523	4	Op524	4	Op525	4
Op526	4	Op527	4	Op528	4
Op529	4	Op530	4	Op531	4
Op532	4	Op533	4	Op534	4
Op535	4	Op536	4	Op537	4
Op538	4	Op539	4	Op540	4
Op541	4	Op542	4	Op543	4
Op544	4	Op545	4	Op546	4
Op547	4	Op548	4	Op549	4
Op550	4	Op551	4	Op552	4
Op553	4	Op554	4	Op555	4
Op556	4	Op557	4	Op558	4
Op559	4	Op560	4	Op561	4
Op562	4	Op563	4	Op564	4
Op565	4	Op566	4	Op567	4
Op568	4	Op569	4	Op570	4
Op571	4	Op572	4	Op573	4
Op574	4	Op575	4	Op576	4
Op577	4	Op578	4	Op579	4
Op580	4	Op581	4	Op582	4
Op583	4	Op584	4	Op585	4
Op586	4	Op587	4	Op588	4
Op589	4	Op590	4	Op591	4
Op592	4	Op593	4	Op594	4
Op595	4	Op596	4	Op597	4
Op598	4	Op599	4	Op600	4
Op601	4	Op602	4	Op603	4
Op604	4	Op605	4	Op606	4
Op607	4	Op608	4	Op609	4
Op610	4	Op611	4	Op612	4
Op613	4	Op614	4	Op615	4
Op616	4	Op617	4	Op618	4
Op619	4	Op620	4	Op621	4
Op622	4	Op623	4	Op624	4
Op625	4	Op626	4	Op627	4
Op628	4	Op629	4	Op630	4
Op631	4	Op632	4	Op633	4
Op634	4	Op635	4	Op636	4
Op637	4	Op638	4	Op639	4
Op640	4	Op641	4	Op642	4
Op643	4	Op644	4	Op645	4
Op646	4	Op647	4	Op648	4
Op649	4	Op650	4	Op651	4
Op652	4	Op653	4	Op654	4
Op655	4	Op656	4	Op657	4
Op658	4	Op659	4	Op660	4
Op661	4	Op662	4	Op663	4
Op664	4	Op665	4	Op666	4
Op667	4	Op668	4	Op669	4
Op670	4	Op671	4	Op672	4
Op673	4	Op674	4	Op675	4
Op676	4	Op677	4	Op678	4
Op679	4	Op680	4	Op681	4
Op682	4	Op683	4	Op684	4
Op685	4	Op686	4	Op687	4
Op688	4	Op689	4	Op690	4
Op691	4	Op692	4	Op693	4
Op694	4	Op695	4	Op696	4
Op697	4	Op698	4	Op699	4
Op700	4	Op701	4	Op702	4
Op703	4	Op704	4	Op705	4
Op706	4	Op707	4	Op708	4
Op709	4	Op710	4	Op711	4
Op712	4	Op713	4	Op714	4
Op715	4	Op716	4	Op717	4
Op718	4	Op719	4	Op720	4
Op721	4				

Approach #1: long-hand multiplication

Example

Consider a case where $b = 10$, $x = 623_{(10)}$ and $y = 567_{(10)}$:

$x =$	$623_{(10)} \mapsto$			6	2	3		
$y =$	$567_{(10)} \mapsto$			5	6	7	\times	
$p_0 = 7 \cdot 3 \cdot 10^0 =$	$21_{(10)} \mapsto$				2	1		
$p_1 = 7 \cdot 2 \cdot 10^1 =$	$140_{(10)} \mapsto$				1	4		
$p_2 = 7 \cdot 6 \cdot 10^2 =$	$4200_{(10)} \mapsto$		4		2			
$p_3 = 6 \cdot 3 \cdot 10^1 =$	$180_{(10)} \mapsto$				1	8		
$p_4 = 6 \cdot 2 \cdot 10^2 =$	$1200_{(10)} \mapsto$			1	2			
$p_5 = 6 \cdot 6 \cdot 10^3 =$	$36000_{(10)} \mapsto$		3		6			
$p_6 = 5 \cdot 3 \cdot 10^2 =$	$1500_{(10)} \mapsto$				1	5		
$p_7 = 5 \cdot 2 \cdot 10^3 =$	$10000_{(10)} \mapsto$			1	0			
$p_8 = 5 \cdot 6 \cdot 10^4 =$	$300000_{(10)} \mapsto$		3		0			
$r =$	$353241_{(10)} \mapsto$		3	5	3	2	4	1

Notes:

Approach #1: long-hand multiplication

Example (operand scanning)

Consider an example where where $|x| = |y| = 3$:

The diagram illustrates the computation of the product of two polynomials using the Karatsuba algorithm. The input polynomials are X_2, X_1, X_0 and y_2, y_1, y_0 . The intermediate products are $y_0 \cdot X_0$, $y_1 \cdot X_0$, $y_0 \cdot X_1$, $y_1 \cdot X_1$, $y_0 \cdot X_2$, $y_1 \cdot X_2$, and $y_2 \cdot X_2$. The final result is shown as a polynomial with coefficients r_5, r_4, r_3, r_2, r_1 , and r_0 .

Notice that

1. an outer-loop steps through digits of y , say y_i ,
2. an inner-loop steps through digits of x , say x_j .

Example (product scanning)

Consider an example where where $|x| = |y| = 3$:

The diagram shows the multiplication of two polynomials, (x_2, x_1, x_0) and (y_2, y_1, y_0) . The partial products are arranged in a triangular fashion, with each row shifted to the right relative to the previous one. The partial products are:

- Row 1: $y_0 \cdot x_0$
- Row 2: $y_1 \cdot x_0$ and $y_0 \cdot x_1$
- Row 3: $y_2 \cdot x_0$, $y_1 \cdot x_1$, and $y_0 \cdot x_2$
- Row 4: $y_2 \cdot x_1$, $y_1 \cdot x_2$
- Row 5: $y_2 \cdot x_2$

These partial products are then summed to produce the final result, which is shown in a table at the bottom with columns labeled $r_5, r_4, r_3, r_2, r_1, r_0$.

Notice that

1. an outer-loop steps through digits of r , say r_i ,
2. two inner-loops step through matching digits of x and y , say x_j and x_i .

Notes:

Algorithm (operand scanning)

Input: Two unsigned, base- b integers x and y
Output: An unsigned, base- b integer $r = x \cdot y$

```
1  $l_x \leftarrow |x|, l_y \leftarrow |y|, l_r \leftarrow l_x + l_y$ 
2  $r \leftarrow 0$ 
3 for  $j = 0$  upto  $l_y - 1$  step  $+1$  do
4    $c \leftarrow 0$ 
5   for  $i = 0$  upto  $l_x - 1$  step  $+1$  do
6      $u \cdot b + v = t \leftarrow y_j \cdot x_i + r_{j+i} + c$ 
7      $r_{j+i} \leftarrow v$ 
8      $c \leftarrow u$ 
9   end
10   $r_{j+l_x} \leftarrow c$ 
11 end
12 return  $r$ 
```

Notes:

Example (operand scanning)

Consider a case where $b = 10$, $x = 623_{(10)}$ and $y = 567_{(10)}$:

j	i	r	c	y_i	x_j	$t = y_i \cdot x_i + r_{i+j} + c$	r'	c'
0	0	$\langle 0, 0, 0, 0, 0, 0 \rangle$	0	7	3	21	$\langle 1, 0, 0, 0, 0, 0 \rangle$	2
0	1	$\langle 1, 0, 0, 0, 0, 0 \rangle$	2	7	2	16	$\langle 1, 6, 0, 0, 0, 0 \rangle$	1
0	2	$\langle 1, 6, 0, 0, 0, 0 \rangle$	1	7	6	43	$\langle 1, 6, 3, 0, 0, 0 \rangle$	4
0		$\langle 1, 6, 3, 0, 0, 0 \rangle$	4				$\langle 1, 6, 3, 4, 0, 0 \rangle$	
1	0	$\langle 1, 6, 3, 4, 0, 0 \rangle$	0	6	3	24	$\langle 1, 4, 3, 4, 0, 0 \rangle$	2
1	1	$\langle 1, 4, 3, 4, 0, 0 \rangle$	2	6	2	17	$\langle 1, 4, 7, 4, 0, 0 \rangle$	1
1	2	$\langle 1, 4, 7, 4, 0, 0 \rangle$	1	6	6	41	$\langle 1, 4, 7, 1, 0, 0 \rangle$	4
1		$\langle 1, 4, 7, 1, 0, 0 \rangle$	4				$\langle 1, 4, 7, 1, 4, 0 \rangle$	
2	0	$\langle 1, 4, 7, 1, 4, 0 \rangle$	0	5	3	22	$\langle 1, 4, 2, 1, 4, 0 \rangle$	2
2	1	$\langle 1, 4, 2, 1, 4, 0 \rangle$	2	5	2	13	$\langle 1, 4, 2, 3, 4, 0 \rangle$	1
2	2	$\langle 1, 4, 2, 3, 5, 0 \rangle$	1	5	6	35	$\langle 1, 4, 2, 3, 5, 0 \rangle$	3
2		$\langle 1, 4, 2, 3, 5, 0 \rangle$	3				$\langle 1, 4, 2, 3, 5, 3 \rangle$	3

Notes:

Algorithm (product scanning)

```
Input: Two unsigned, base- $b$  integers  $x$  and  $y$   
Output: An unsigned, base- $b$  integer  $r = x \cdot y$   
1  $l_x \leftarrow |x|, l_y \leftarrow |y|, l_r \leftarrow l_x + l_y$   
2  $r \leftarrow 0, c_0 \leftarrow 0, c_1 \leftarrow 0, c_2 \leftarrow 0$   
3 for  $k = 0$  upto  $l_x + l_y - 1$  step  $+1$  do  
4   for  $j = 0$  upto  $l_y - 1$  step  $+1$  do  
5     for  $i = 0$  upto  $l_x - 1$  step  $+1$  do  
6       if  $(j + i) = k$  then  
7          $u \cdot b + v = t \leftarrow y_j \cdot x_i$   
8          $c \cdot b + c_0 = t \leftarrow c_0 + v$   
9          $c \cdot b + c_1 = t \leftarrow c_1 + u + c$   
10         $c_2 \leftarrow c_2 + c$   
11      end  
12    end  
13  end  
14   $r_k \leftarrow c_0, c_0 \leftarrow c_1, c_1 \leftarrow c_2, c_2 \leftarrow 0$   
15 end  
16  $r_{l_x+l_y-1} \leftarrow c_0$ 
```

Notes:

Example (product scanning)

Consider a case where $b = 10, x = 623_{(10)}$ and $y = 567_{(10)}$:

k	j	i	r	c_2	c_1	c_0	y_i	x_j	$t = y_i \cdot x_j$	r'	c'_2	c'_1	c'_0
0	0	0	$\langle 0, 0, 0, 0, 0, 0 \rangle$	0	0	0				$\langle 0, 0, 0, 0, 0, 0 \rangle$	0	2	1
0			$\langle 0, 0, 0, 0, 0, 0 \rangle$	0	0	0	7	3	21	$\langle 1, 0, 0, 0, 0, 0 \rangle$	0	0	2
1	0	1	$\langle 1, 0, 0, 0, 0, 0 \rangle$	0	0	2	7	2	14	$\langle 1, 0, 0, 0, 0, 0 \rangle$	0	1	6
1	1	0	$\langle 1, 0, 0, 0, 0, 0 \rangle$	0	1	6	6	3	18	$\langle 1, 0, 0, 0, 0, 0 \rangle$	0	3	4
1			$\langle 1, 0, 0, 0, 0, 0 \rangle$	0	3	4				$\langle 1, 4, 0, 0, 0, 0 \rangle$	0	0	3
2	0	2	$\langle 1, 4, 0, 0, 0, 0 \rangle$	0	0	3	7	6	42	$\langle 1, 4, 0, 0, 0, 0 \rangle$	0	4	5
2	1	1	$\langle 1, 4, 0, 0, 0, 0 \rangle$	0	4	5	6	2	12	$\langle 1, 4, 0, 0, 0, 0 \rangle$	0	5	7
2	2	0	$\langle 1, 4, 0, 0, 0, 0 \rangle$	0	4	7	5	3	15	$\langle 1, 4, 0, 0, 0, 0 \rangle$	0	7	2
2			$\langle 1, 4, 0, 0, 0, 0 \rangle$	0	7	2				$\langle 1, 4, 2, 0, 0, 0 \rangle$	0	0	7
3	1	2	$\langle 1, 4, 2, 0, 0, 0 \rangle$	0	0	7	6	6	36	$\langle 1, 4, 2, 0, 0, 0 \rangle$	0	4	3
3	2	1	$\langle 1, 4, 2, 0, 0, 0 \rangle$	0	4	3	5	2	10	$\langle 1, 4, 2, 0, 0, 0 \rangle$	0	5	3
3			$\langle 1, 4, 2, 0, 0, 0 \rangle$	0	5	3				$\langle 1, 4, 2, 3, 0, 0 \rangle$	0	0	5
4	2	2	$\langle 1, 4, 2, 3, 0, 0 \rangle$	0	0	5	5	6	30	$\langle 1, 4, 2, 3, 0, 0 \rangle$	0	3	5
4			$\langle 1, 4, 2, 3, 0, 0 \rangle$	0	3	5				$\langle 1, 4, 2, 3, 5, 0 \rangle$	0	0	3
			$\langle 1, 4, 2, 3, 5, 0 \rangle$	0	0	3				$\langle 1, 4, 2, 3, 5, 3 \rangle$	0	0	3
			$\langle 1, 4, 2, 3, 5, 3 \rangle$										

Notes:

- **Idea:** scalar multiplication of x by y is simply repeated addition, i.e.,

$$y \cdot x = \underbrace{x + x + \cdots + x + x}_{y \text{ terms}},$$

so, if $y = 14_{(10)}$, then

$$14 \cdot x = x + x + x + x + x + x + x + x + x + x + x + x + x + x.$$

Notes:

- **Idea:** by writing y in base- b we express the product as a summation, i.e.,

$$y \cdot x \equiv \left(\sum_{i=0}^{n-1} y_i \cdot b^i \right) \cdot x \equiv \sum_{i=0}^{n-1} y_i \cdot x \cdot b^i$$

so, if $y = 14_{(10)} \mapsto 1110_{(2)}$, then

$$\begin{aligned} y \cdot x &= y_0 \cdot x \cdot 2^0 + y_1 \cdot x \cdot 2^1 + y_2 \cdot x \cdot 2^2 + y_3 \cdot x \cdot 2^3 \\ &= 0 \cdot x \cdot 2^0 + 1 \cdot x \cdot 2^1 + 1 \cdot x \cdot 2^2 + 1 \cdot x \cdot 2^3 \\ &= 0 \cdot x + 2 \cdot x + 4 \cdot x + 8 \cdot x \\ &= 14 \cdot x \end{aligned}$$

noting that *if* $b = 2$, for any t

1. we know

$$t \cdot 2^i \equiv t \ll i,$$

and

2. we know

$$t \cdot y_i \equiv \begin{cases} t & \text{if } y_i = 1 \\ 0 & \text{if } y_i = 0 \end{cases}$$

Notes:

- Idea: for $b = 2$ and given

$$\sum_{i=0}^{n-1} y_i \cdot x \cdot 2^i,$$

construct a combinatorial design to compute r via

1. some AND gates to generate partial products (i.e., $y_i \cdot x$),
2. some left-shift components to scale the partial products correctly (i.e., $y_i \cdot x \cdot 2^i$), and
3. some adder components to sum the scaled partial products

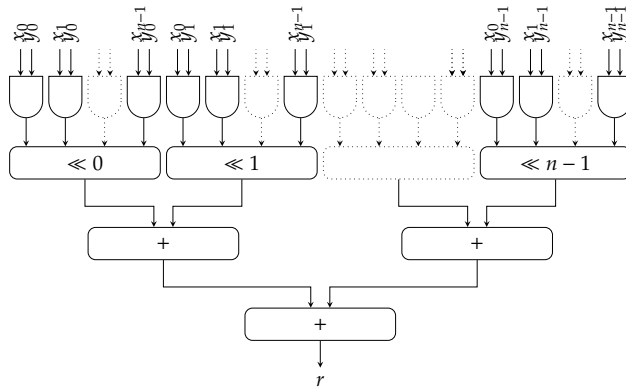
i.e., ...

Notes:

Multiplication in practice (2)

Candidate design #2: combinatorial, bit-parallel

Circuit



- Note that this design

- ve: requires a larger data-path
- +ve: requires a smaller control-path (i.e., none at all),
- +ve: requires less steps (i.e., 1),
- ve: has a longer critical path (meaning each step is longer).

Notes:

► Idea: for $b = 2$, apply **Horner's rule** st. if $y = 14_{(10)} \mapsto 1110_{(2)}$ we (re)bracket

$$\sum_{i=0}^{n-1} y_i \cdot x \cdot 2^i$$

as

$$\begin{aligned} y \cdot x &= y_0 \cdot x + 2 \cdot (y_1 \cdot x + 2 \cdot (y_2 \cdot x + 2 \cdot (y_3 \cdot x + 2 \cdot (0)))) \\ &= 0 \cdot x + 2 \cdot (1 \cdot x + 2 \cdot (1 \cdot x + 2 \cdot (1 \cdot x + 2 \cdot (0)))) \\ &= 0 \cdot x + 2 \cdot (1 \cdot x + 2 \cdot (1 \cdot x + 2 \cdot (1 \cdot x + 0))) \\ &= 0 \cdot x + 2 \cdot (1 \cdot x + 2 \cdot (1 \cdot x + 2 \cdot (1 \cdot x))) \\ &= 0 \cdot x + 2 \cdot (1 \cdot x + 2 \cdot (1 \cdot x + 2 \cdot x)) \\ &= 0 \cdot x + 2 \cdot (1 \cdot x + 2 \cdot (3 \cdot x)) \\ &= 0 \cdot x + 2 \cdot (1 \cdot x + 6 \cdot x) \\ &= 0 \cdot x + 2 \cdot (7 \cdot x) \\ &= 0 \cdot x + 14 \cdot x \\ &= 14 \cdot x \end{aligned}$$

then ...

Notes:

► ... evaluate the expression inside-out, *accumulating* the result via the rule

$$r \leftarrow y_i \cdot x + 2 \cdot r$$

i.e.,

Algorithm

Input: Two unsigned, n -bit, base-2 integers x and y
Output: An unsigned, $2n$ -bit, base-2 integer $r = y \cdot x$

```
1  $r \leftarrow 0$ 
2 for  $i = n - 1$  downto 0 step  $-1$  do
3    $r \leftarrow 2 \cdot r$ 
4   if  $y_i = 1$  then
5      $r \leftarrow r + x$ 
6   end
7 end
8 return  $r$ 
```

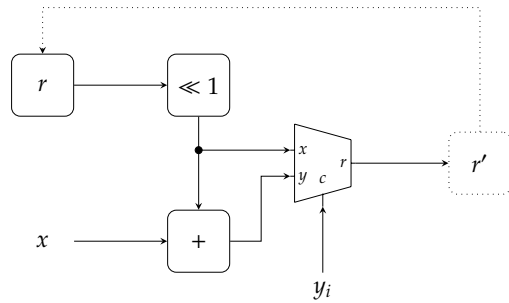
Example

Consider a case where $y = 14_{(10)} \mapsto 1110_{(2)}$:

i	r	y_i	r'	
	0			
3	0	1	x	$r' \leftarrow 2 \cdot r + x$
2	x	1	$3 \cdot x$	$r' \leftarrow 2 \cdot r + x$
1	$3 \cdot x$	1	$7 \cdot x$	$r' \leftarrow 2 \cdot r + x$
0	$7 \cdot x$	0	$14 \cdot x$	$r' \leftarrow 2 \cdot r$

Notes:

Circuit



► Note that this design

- +ve: requires a smaller data-path
- ve: requires a larger control-path (i.e., an entire FSM),
- ve: requires more steps (i.e., n),
- +ve: has a shorter critical path (meaning each step is shorter).

Conclusions

► Take away points:

1. The design strategy we used is important and (fairly) general-purpose:
 - 1.1 explore and understand an approach in theory,
 - 1.2 translate, formalise, and generalise the approach into an algorithm,
 - 1.3 translate the algorithm into a hardware design,
 - 1.4 refine (or select) the hardware design to satisfy any design constraints.
2. Computer arithmetic is an interesting special-case:
 - it's a broad topic with a rich history,
 - there's usually a large design space of potential approaches,
 - they're often easy to understand at an intuitive, high level,
 - correctness and efficiency of resulting low-level solutions is vital and challenging.

Notes:

Notes:

Additional Reading

- ▶ [Wikipedia: Computer Arithmetic](http://en.wikipedia.org/wiki/Category:Computer_arithmetic). URL: http://en.wikipedia.org/wiki/Category:Computer_arithmetic.
- ▶ D. Page. “Chapter 7: Arithmetic and logic”. In: *A Practical Introduction to Computer Architecture*. 1st ed. Springer-Verlag, 2009.
- ▶ B. Parhami. “Part 3: Multiplication”. In: *Computer Arithmetic: Algorithms and Hardware Designs*. 1st ed. Oxford University Press, 2000.
- ▶ W. Stallings. “Chapter 10: Computer arithmetic”. In: *Computer Organisation and Architecture*. 9th ed. Prentice-Hall, 2013.
- ▶ A.S. Tanenbaum and T. Austin. “Section 3.2.2: Arithmetic circuits”. In: *Structured Computer Organisation*. 6th ed. Prentice-Hall, 2012.

Notes:

References

- [1] [Wikipedia: Computer Arithmetic](http://en.wikipedia.org/wiki/Category:Computer_arithmetic). URL: http://en.wikipedia.org/wiki/Category:Computer_arithmetic (see p. 37).
- [2] D. Page. “Chapter 7: Arithmetic and logic”. In: *A Practical Introduction to Computer Architecture*. 1st ed. Springer-Verlag, 2009 (see p. 37).
- [3] B. Parhami. “Part 3: Multiplication”. In: *Computer Arithmetic: Algorithms and Hardware Designs*. 1st ed. Oxford University Press, 2000 (see p. 37).
- [4] W. Stallings. “Chapter 10: Computer arithmetic”. In: *Computer Organisation and Architecture*. 9th ed. Prentice-Hall, 2013 (see p. 37).
- [5] A.S. Tanenbaum and T. Austin. “Section 3.2.2: Arithmetic circuits”. In: *Structured Computer Organisation*. 6th ed. Prentice-Hall, 2012 (see p. 37).

Notes: