

コンピュータシステムの 理論と実装

2025/2/23

1章 ハードウェア実装

最初 NAND ゲートのみから与えられる。

NAND ゲートから各ゲートを実装していく。

NAND (No + And)

a	b	out
0	0	1
1	0	1
0	1	1
1	1	0

NOT

a	out
0	1
1	0
0	1
1	0

OR

a	b	out
0	0	0
1	0	1
0	1	1
1	1	1

AND

a	b	out
0	0	0
1	0	0
0	1	0
1	1	1

✓ Xor (And, Or, Not)

a	b	out
0	0	0
1	0	1
0	1	1
1	1	0

a, b が異なる時にビットが立つ
a, b が同じ時にビットは立たない。

Xor.hdl ファイルの if 文を見て解いてしまったので、
表から if 文作って hdl 文作成しようか
勉強にしよう

Mux (マルチプレクサ)

sel	out
0	a
1	b

if sel == 0 out = a
else out = b

$a \text{ Or } sel = out$ $(a \text{ Or } sel) \text{ Or } (b \text{ And } sel)$

0	0	0	0
0	0	1	0
1	0	0	0
1	0	1	0

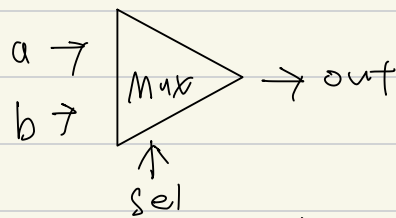
$b \text{ And } sel = out$

Max

$(a \text{ Or } sel) \text{ Or } (\text{Not } b \text{ And } sel)$

0	0	1	1	0	0	1
0	0	1	0	0	0	1
1	0	1	1	0	0	1
1	0	1	0	0	0	1
			Or			
	0	1			0	1
	0	1			0	0
	1	1			0	1
	1	1			0	0

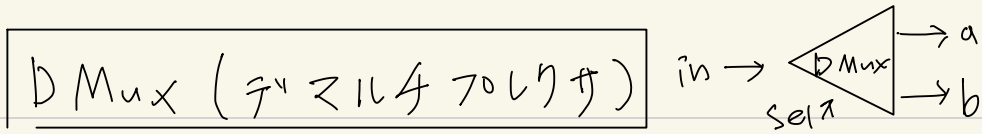
	Or
0	1
0	1
1	1
1	1



sel	out
0	a
1	b

	Xor
0	0
0	1
1	0
1	1

$(a \text{ Or } sel) \text{ Xor } (\text{Not } b \text{ And } sel)$



sel	a	b	in, sel	$if (sel == 0)$
0	in	0	a, b	$\{a, b\} = \{in, 0\}$
1	0	in		else $\{a, b\} = \{0, in\}$

in	sel	a	Xor	And	or	NAnd
0	0	0	0	0	0	1
1	0	1	1	0	1	1
0	1	0	1	0	1	1
1	1	0	0	1	1	0

in	sel	b
0	0	0
1	0	0
0	1	0
1	1	1

$outb = And(a = in, b = sel, out = b)$

$outa = Xor(a = in, b = outb, out = a)$

ifDL:

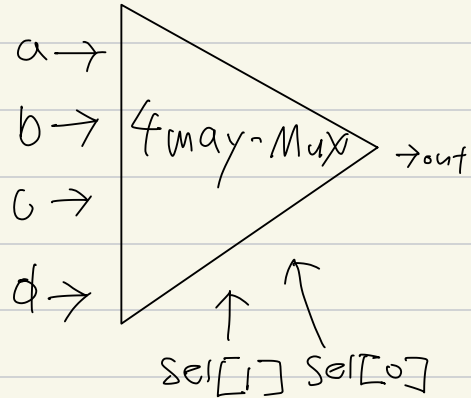
$And(a = in, b = sel, out = outb, out = b);$

$Xor(a = in, b = outb, out = a);$

P. 388 ~~48~~ ~~48~~

Mux 4Way lb

sel[1]	sel[0]	out
0	0	a
0	1	b
1	0	c
1	1	d



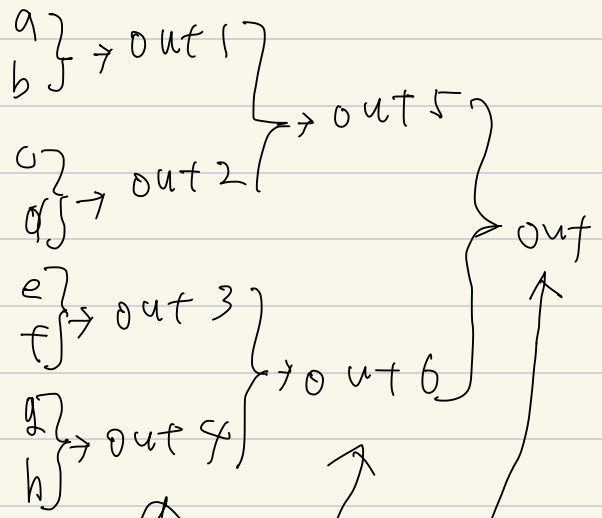
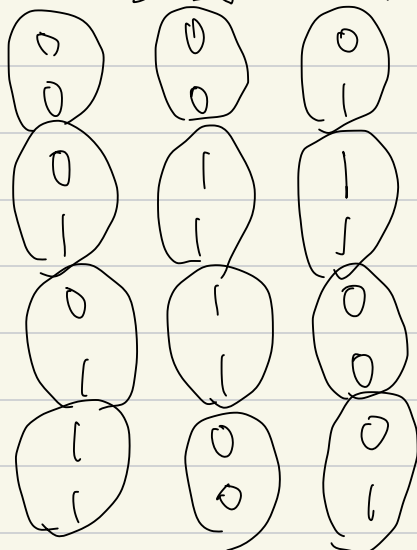
a	b	c	d	sel[1]	sel[0]	out
1	0	0	0	0	0	1
1	1	0	0	0	1	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

sel[1]が0のは a, b を出力, sel[1]が1のは c, d を出力

$\text{Mux lb}(a=a, b=b, \text{sel}=\text{sel}[0], \text{out}=\text{out}_0); \leftarrow \text{sel}[1]=0$
 $\text{Mux lb}(a=c, b=d, \text{sel}=\text{sel}[0], \text{out}=\text{out}_1); \leftarrow \text{sel}[1]=1$
 $\text{Mux lb}(a=\text{out}_0, b=\text{out}_1, \text{sel}=\text{sel}[1], \text{out}=\text{out});$

Mux 8 Way 1b

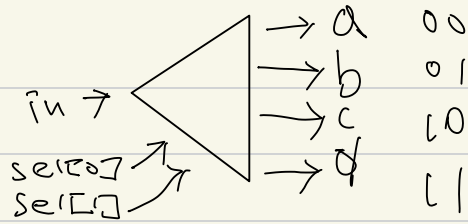
sel0 sel1 sel2



Mux 4 Way 1b

Mux 1b

DMax 4Way



in	sel[0]	sel[1]	a	b	c	d
0	0	0	0	0	0	0
1	0	1	0	1	0	0
0	1	0	0	0	0	0
1	1	1	0	0	0	1
1	0	0	1	0	0	0
0	0	1	0	0	0	0
1	1	0	0	0	1	0
0	1	1	0	0	0	0

sel[0] が 0 のときは c, d を 0 に置き、1 のときは a-b を出す。

DMax (in=in, sel=sel[0], a=out a, b=out b);

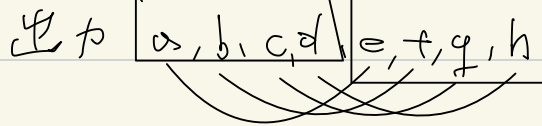
~~DMax (in=in, sel=sel[1], a=out c, b=out d);~~

DMax (in=out a, sel=sel[1], a=a, b=c);

DMax (in=out b, sel=sel[1], a=b, b=d);

DMux 8Way

入力 in , $sel[3]$



- $sel[2]$ 出力 0 のときは e, f, g, h 出力 1 になる
- $sel[2]$ 出力 1 のときは a, b, c, d 出力 0 になる

DMux 4Way ($in=in$, $sel=sel[0..1]$, $a=a$, $b=b$, $c=c$, $d=d$);

DMux 4Way ($in=in$, $sel=sel[0..1]$, $a=e$, $b=f$, $c=g$, $d=h$);

DMux ($in=a$, $sel=sel[2]$, $a=a$, $b=e$);

DMux ($in=b$, $sel=sel[2]$, $a=b$, $b=f$);

DMux ($in=c$, $sel=sel[2]$, $a=c$, $b=g$);

DMux ($in=d$, $sel=sel[2]$, $a=d$, $b=h$);

2025/2/27 2章 7-10 算術

半加算器 (Half Adder)

入力 a, b

a	b	carry	sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

出力 Sum, Carry
 $a+b$ の
最下位ビット
最上位ビット

Xor ($a=a, b=b, \text{out}=\text{sum}$);

And ($a=a, b=b, \text{out}=\text{carry}$);

全加器 (Full Adder)

输入 a, b, c

输出 $carry, sum$

a	b	$carry$	sum	$c + sum$	$c\ carry$	$csum$
0	0	0	0	0	0	0
0	1	0	1	0	1	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0
0	0	0	0	1	0	1
0	1	0	1	1	1	0
1	0	0	1	1	1	0
1	1	1	0	1	0	1

$abcarry \text{ or } ccarry = carry$

$csum = sum$

Half Adder ($a = a, b = b, carry = abcarry, sum = absum$);
 Half Adder ($a = c, b = absum, carry = ccarry, sum = sum$);
 Or ($a = abcarry, b = ccarry, out = carry$);

16ビット加算器 (Add16)

2ビットの場合

10 a

01 + b

11 out

11

11 a

11 + b

110 out

入力 a[16], b[16]

出力 out[16]

Overflow (入力ビット幅を超え) は無視

Half Adder (a = a[0], b = b[0], carry = 0, sum = out[0])

Full Adder (a = a[i], b = b[i], c = 0, carry = 0, sum = out[i])

i: ビット繰り返し

a[0] = 1, b[0] = 1, c = 0, out[0] = 0 i = 0

a[1] = 1, b[1] = 1, c = 1, out[1] = 1 i = 1

$\lambda \in \text{In}[6]$

出力 6 桁 [16]

関数 $out = in + 1$

・1つ以上のタプルを持つための1が必要 \rightarrow true p374

- ・ 最下位ビット + 1 は XOR で求めらる? $\rightarrow \text{sum} = \text{out}[i]$

0 空集の上位が1発生するかは $A \cap b \neq \emptyset$ である。 \rightarrow Carby

$Xor(a = \text{true}, b = \text{in}[0], out = out[0]);$
 $And(a = \text{in}, b = \text{in}[0], out = carry0);$
 $Xor(a = carry0, b = \text{in}[1], out = out[1]);$
 $And(a = carry0, b = \text{in}[1], out = carry1);$

この方法で実装すると $(Xor \times 16) + (And \times 15)$ の構成となる
 Add16.hdl を利用

carry1 = carry0 And 2nd 3rd 4th 5th 6th 7th 8th 9th 10th 11th 12th 13th 14th 15th 16th 17th 18th 19th 20th 21st 22nd 23rd 24th 25th 26th 27th 28th 29th 30th 31st 32nd 33rd 34th 35th 36th 37th 38th 39th 40th 41st 42nd 43rd 44th 45th 46th 47th 48th 49th 50th 51st 52nd 53rd 54th 55th 56th 57th 58th 59th 60th 61st 62nd 63rd 64th 65th 66th 67th 68th 69th 70th 71st 72nd 73rd 74th 75th 76th 77th 78th 79th 80th 81st 82nd 83rd 84th 85th 86th 87th 88th 89th 90th 91st 92nd 93rd 94th 95th 96th 97th 98th 99th 100th 101st 102nd 103rd 104th 105th 106th 107th 108th 109th 110th 111th 112th 113th 114th 115th 116th 117th 118th 119th 120th 121st 122nd 123rd 124th 125th 126th 127th 128th 129th 130th 131st 132nd 133rd 134th 135th 136th 137th 138th 139th 140th 141st 142nd 143rd 144th 145th 146th 147th 148th 149th 150th 151st 152nd 153rd 154th 155th 156th 157th 158th 159th 160th 161st 162nd 163rd 164th 165th 166th 167th 168th 169th 170th 171st 172nd 173rd 174th 175th 176th 177th 178th 179th 180th 181st 182nd 183rd 184th 185th 186th 187th 188th 189th 190th 191st 192nd 193rd 194th 195th 196th 197th 198th 199th 200th 201st 202nd 203rd 204th 205th 206th 207th 208th 209th 210th 211th 212th 213th 214th 215th 216th 217th 218th 219th 220th 221st 222nd 223rd 224th 225th 226th 227th 228th 229th 230th 231st 232nd 233rd 234th 235th 236th 237th 238th 239th 240th 241st 242nd 243rd 244th 245th 246th 247th 248th 249th 250th 251st 252nd 253rd 254th 255th 256th 257th 258th 259th 260th 261st 262nd 263rd 264th 265th 266th 267th 268th 269th 270th 271st 272nd 273rd 274th 275th 276th 277th 278th 279th 280th 281st 282nd 283rd 284th 285th 286th 287th 288th 289th 290th 291st 292nd 293rd 294th 295th 296th 297th 298th 299th 300th 301st 302nd 303rd 304th 305th 306th 307th 308th 309th 310th 311th 312th 313th 314th 315th 316th 317th 318th 319th 320th 321st 322nd 323rd 324th 325th 326th 327th 328th 329th 330th 331st 332nd 333rd 334th 335th 336th 337th 338th 339th 340th 341st 342nd 343rd 344th 345th 346th 347th 348th 349th 350th 351st 352nd 353rd 354th 355th 356th 357th 358th 359th 360th 361st 362nd 363rd 364th 365th 366th 367th 368th 369th 370th 371st 372nd 373rd 374th 375th 376th 377th 378th 379th 380th 381st 382nd 383rd 384th 385th 386th 387th 388th 389th 390th 391st 392nd 393rd 394th 395th 396th 397th 398th 399th 400th 401st 402nd 403rd 404th 405th 406th 407th 408th 409th 410th 411th 412th 413th 414th 415th 416th 417th 418th 419th 420th 421st 422nd 423rd 424th 425th 426th 427th 428th 429th 430th 431st 432nd 433rd 434th 435th 436th 437th 438th 439th 440th 441st 442nd 443rd 444th 445th 446th 447th 448th 449th 450th 451st 452nd 453rd 454th 455th 456th 457th 458th 459th 460th 461st 462nd 463rd 464th 465th 466th 467th 468th 469th 470th 471st 472nd 473rd 474th 475th 476th 477th 478th 479th 480th 481st 482nd 483rd 484th 485th 486th 487th 488th 489th 490th 491st 492nd 493rd 494th 495th 496th 497th 498th 499th 500th 501st 502nd 503rd 504th 505th 506th 507th 508th 509th 510th 511th 512th 513th 514th 515th 516th 517th 518th 519th 520th 521st 522nd 523rd 524th 525th 526th 527th 528th 529th 530th 531st 532nd 533rd 534th 535th 536th 537th 538th 539th 540th 541st 542nd 543rd 544th 545th 546th 547th 548th 549th 550th 551st 552nd 553rd 554th 555th 556th 557th 558th 559th 560th 561st 562nd 563rd 564th 565th 566th 567th 568th 569th 570th 571st 572nd 573rd 574th 575th 576th 577th 578th 579th 580th 581st 582nd 583rd 584th 585th 586th 587th 588th 589th 590th 591st 592nd 593rd 594th 595th 596th 597th 598th 599th 600th 601st 602nd 603rd 604th 605th 606th 607th 608th 609th 610th 611th 612th 613th 614th 615th 616th 617th 618th 619th 620th 621st 622nd 623rd 624th 625th 626th 627th 628th 629th 630th 631st 632nd 633rd 634th 635th 636th 637th 638th 639th 640th 641st 642nd 643rd 644th 645th 646th 647th 648th 649th 650th 651st 652nd 653rd 654th 655th 656th 657th 658th 659th 660th 661st 662nd 663rd 664th 665th 666th 667th 668th 669th 670th 671st 672nd 673rd 674th 675th 676th 677th 678th 679th 680th 681st 682nd 683rd 684th 685th 686th 687th 688th 689th 690th 691st 692nd 693rd 694th 695th 696th 697th 698th 699th 700th 701st 702nd 703rd 704th 705th 706th 707th 708th 709th 710th 711th 712th 713th 714th 715th 716th 717th 718th 719th 720th 721st 722nd 723rd 724th 725th 726th 727th 728th 729th 730th 731st 732nd 733rd 734th 735th 736th 737th 738th 739th 740th 741st 742nd 743rd 744th 745th 746th 747th 748th 749th 750th 751st 752nd 753rd 754th 755th 756th 757th 758th 759th 760th 761st 762nd 763rd 764th 765th 766th 767th 768th 769th 770th 771st 772nd 773rd 774th 775th 776th 777th 778th 779th 780th 781st 782nd 783rd 784th 785th 786th 787th 788th 789th 790th 791st 792nd 793rd 794th 795th 796th 797th 798th 799th 800th 801st 802nd 803rd 804th 805th 806th 807th 808th 809th 810th 811th 812th 813th 814th 815th 816th 817th 818th 819th 820th 821st 822nd 823rd 824th 825th 826th 827th 828th 829th 830th 831st 832nd 833rd 834th 835th 836th 837th 838th 839th 840

carry 0 = inc And inc [0]

$$\text{out}[0] = \text{inc} \text{ xor } \text{in}[0]$$

$\text{out}[i] = \text{carry} \oplus \text{in}[i]$

加算器 $f_{470}(\text{Add } 16, \text{hd } 1)$ は

$(\text{Half Adder} \times 1) + (\text{Full Adder} \times 15)$ で構築される。

$$(Xor \times 1) + (And \times 1) + (\text{HalfAdder} \times 2) + (Or \times 1) \quad (Xor \times 3) + (And \times 3) + Or$$

$$(Xor + And) \times 2$$

Adq 164y70 を利用する $(X \text{ or } Y) + (X \text{ and } Y)$ + 01 を利用する。